L Number	Hits	Search Text	DB	Time stamp
1	10780	("705").CLAS.	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:17
2	282	(705/9).CCLS.	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:18
3	615	(705/8).CCLS.	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:18
4	455	(705/7).CCLS. scan patents	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:18
5	18	internet near2 postage$2 review patents	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:20
7	27533	deliver$3 near2 (time$2 or schedul$3)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:21
8	705	(deliver$3 near2 (time$2 or schedul$3)) same carrier$2	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:23
10	497	ship$5 near2 manag$5	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:22
12	153	(ship$5 near2 manag$5) near3 (system$2 or method$2)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:23
13 Scan titles & rev. patents	161	((deliver$3 near2 (time$2 or schedul$3)) same carrier$2) and (network$2 or internet)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:42
9	33	(deliver$3 near2 (time$2 or schedul$3)) same carrier$2 same (parcel$2 or package$2)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:23
14	411	ship$5 near2 request$3	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:42
15	42	(ship$5 near2 request$3) and (deliver$3 near2 (date$2 or schedule$2))	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 11:43
17	45	(ship$5 near2 request$3) same (date$2)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 12:00
16	12	(ship$5 near2 request$3) same (deliver$3 near2 (date$2 or schedule$2))	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 12:00
18	85	(ship$5 near2 request$3) and (("705").CLAS.)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 12:19
19	5791	shipping near2 cost$2	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/12/04 12:19

20		59	display$3 near3 (shipping near2 cost$2)	USPAT;
				EPO; JPO;
				DERWENT;
				IBM_TDB
					2003/12/04 12:20

review patents

US006058373A

United States Patent [19]

Blinn et al.

[11] **Patent Number: 6,058,373**

[45] **Date of Patent: May 2, 2000**

[54] **SYSTEM AND METHOD FOR PROCESSING ELECTRONIC ORDER FORMS**

[75] Inventors: **Arnold Blinn**, Bellevue, Wash.; **Michael Ari Cohen**, San Francisco, Calif.; **Michael Lorton**; **Gregory J. Stein**, both of Redmond, Wash.

[73] Assignee: **Microsoft Corporation**, Redmond, Wash.

[21] Appl. No.: **08/732,205**

[22] Filed: **Oct. 16, 1996**

[51] **Int. Cl.**[7] **G08B 9/00**

[52] **U.S. Cl.** **705/26**; 705/22; 705/27; 705/28

[58] **Field of Search** 705/20, 22, 26, 705/27, 28, 29

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,799,156	1/1989	Shavit et al. .	
5,347,632	9/1994	Filepp et al. .	
5,710,887	1/1998	Chelliah et al.	395/226

OTHER PUBLICATIONS

General overview and description of eShop Technology, Internet address: http://www.eshop.com/corp/technology.html. This reference was copied from the Internet and printed around May 1996, although the pages are dated Jan. 1, 1996.

A compilation of press releases of various dates describing features of eShop Technology, Internet address: http://www.eshop.com/corp/press.html. This reference was copied from the Internet and printed around May 1996, although the pages are dated Jan. 1, 1996. Also note dates listed for press releases of Nov. 7, 1995, Dec. 7, 1995 and Jan. 23, 1996.

eShop™ Technology Merchant Manual, Feb. 21, 1996. This document contains proprietary material subject to M.P.E.P. § 724.

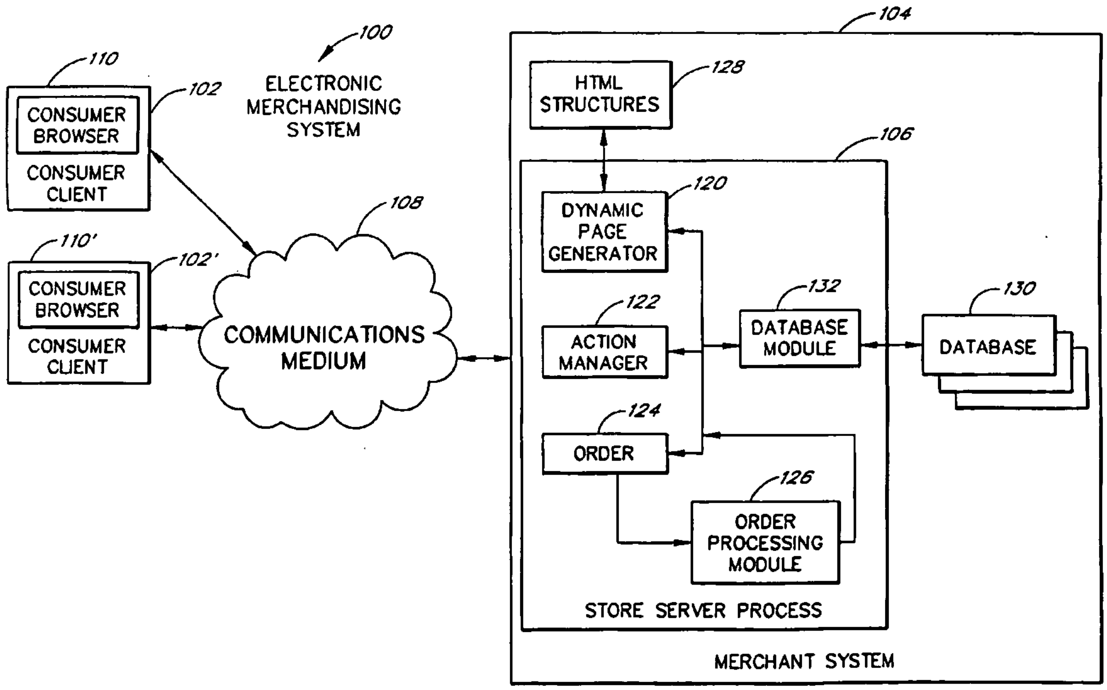
Primary Examiner—Thomas R. Peeso

Attorney, Agent, or Firm—Lee & Hayes, PLLC

[57] **ABSTRACT**

The present invention provides a method and system for processing electronic sales transactions. In a preferred embodiment, an electronic merchandising system allows merchants to create electronic orders which are easily adaptable for different sales situations. The preferred electronic order comprises flexible blackboards which allow merchants to add sales information with what are called key-value pairs. In the preferred embodiment, the order is an object which contains at least one order blackboard and one or more item blackboards. In addition, the preferred embodiment contains an order processing module with multiple stages which process the order. The preferred stages include a product information stage, a merchant information stage, a shopper information stage, an order initialization stage, an order check stage, an item price adjust stage, an order price adjust stage, a shipping stage, a handling stage, a tax stage, an order total stage, an inventory stage, a payment stage and an accept stage.

58 Claims, 18 Drawing Sheets



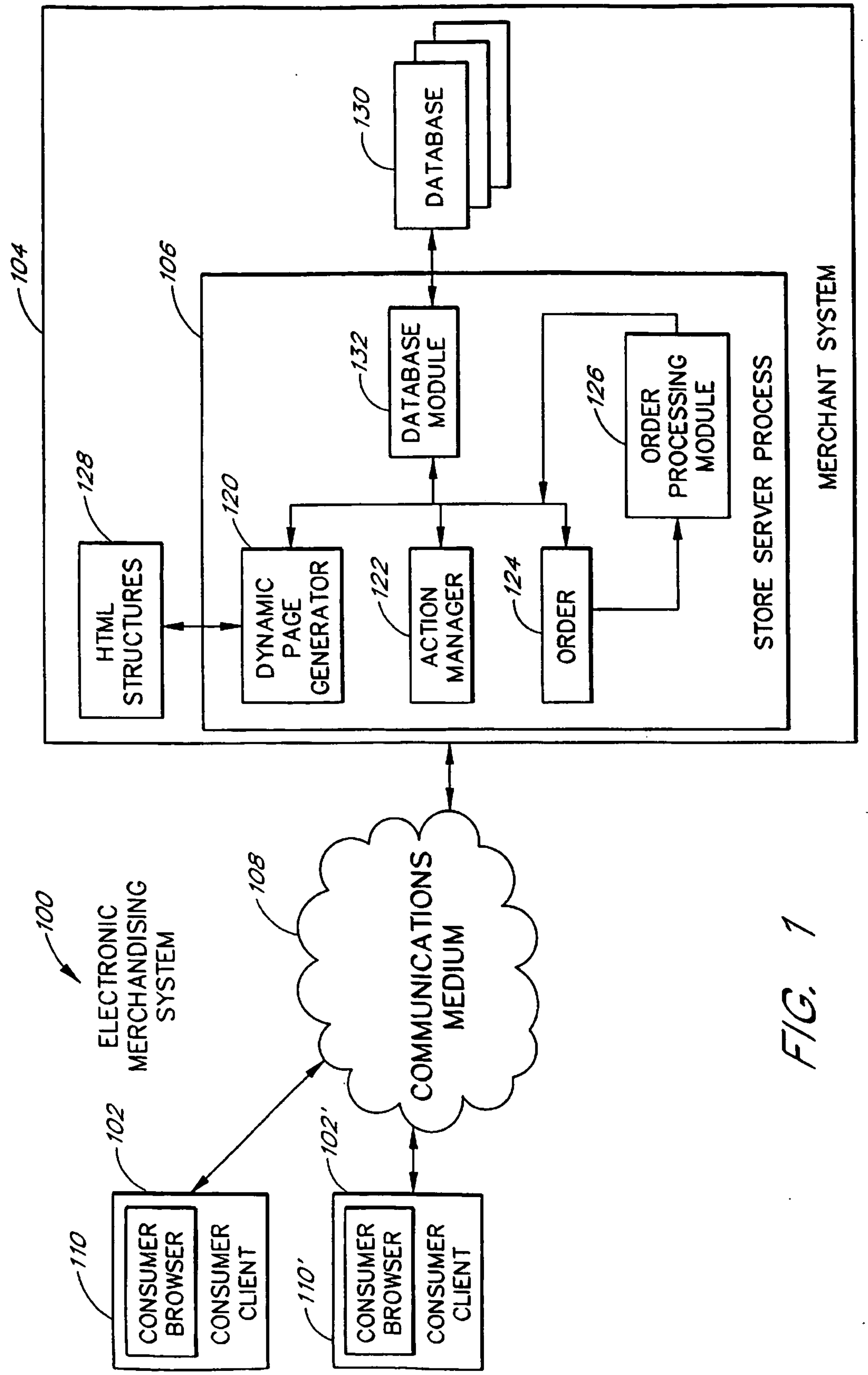

104
106
130
DATABASE
132
DATABASE MODULE
126
ORDER PROCESSING MODULE
128
HTML STRUCTURES
120
DYNAMIC PAGE GENERATOR
122
ACTION MANAGER
124
ORDER
STORE SERVER PROCESS
MERCHANT SYSTEM
100
ELECTRONIC MERCHANDISING SYSTEM
108
COMMUNICATIONS MEDIUM
102
102'
110
CONSUMER BROWSER
CONSUMER CLIENT
110'
CONSUMER BROWSER
CONSUMER CLIENT

FIG. 1

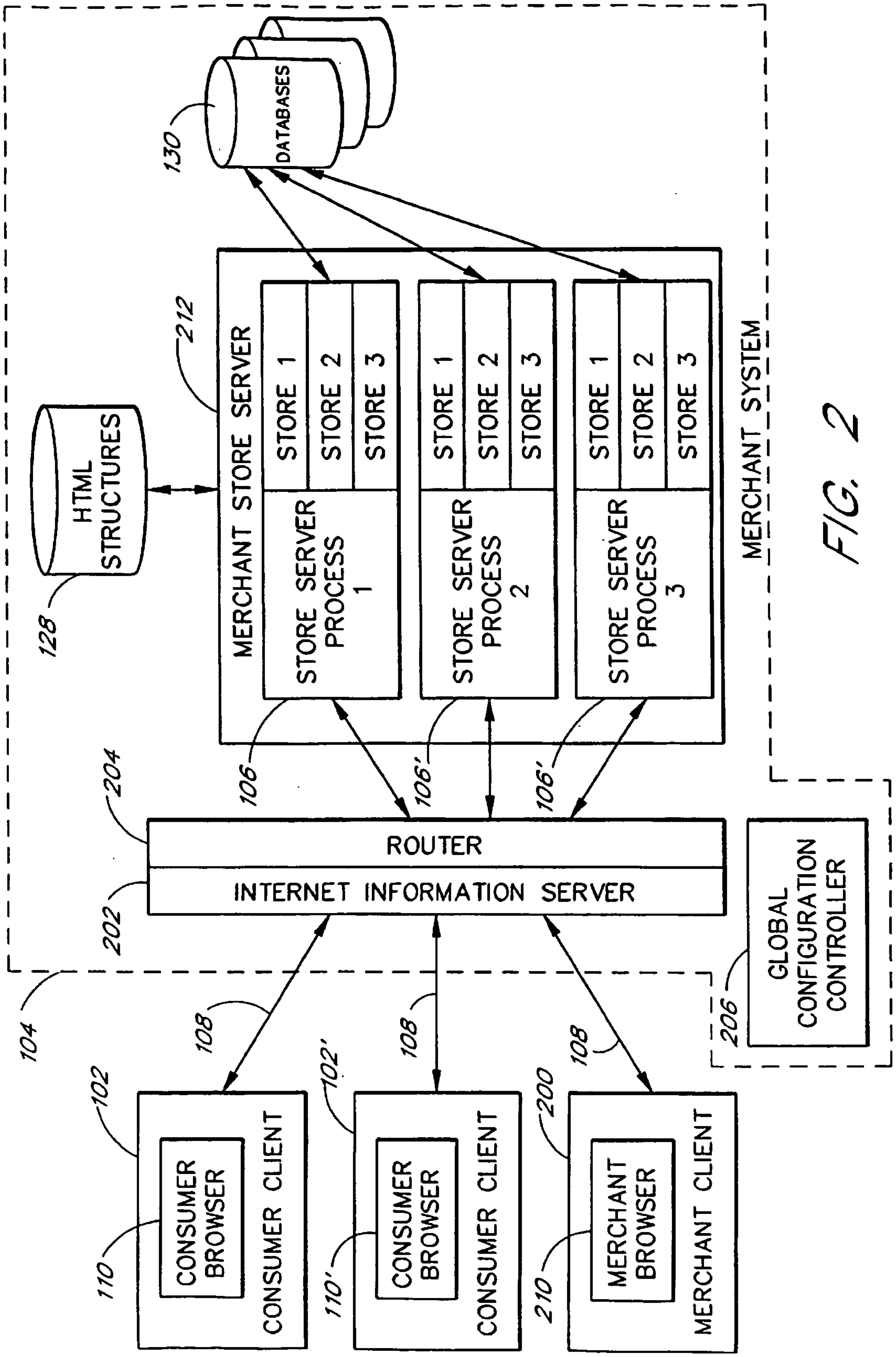
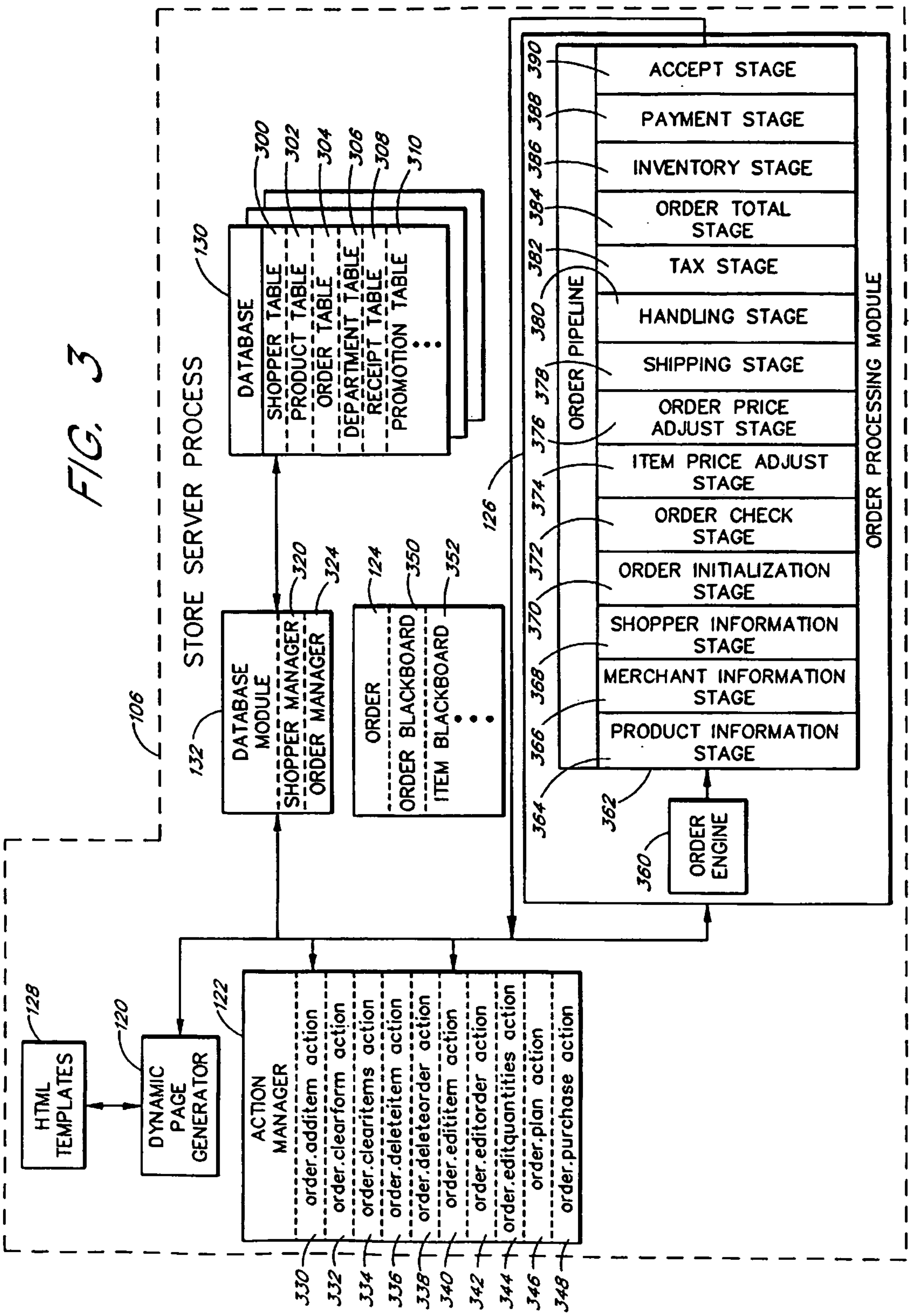
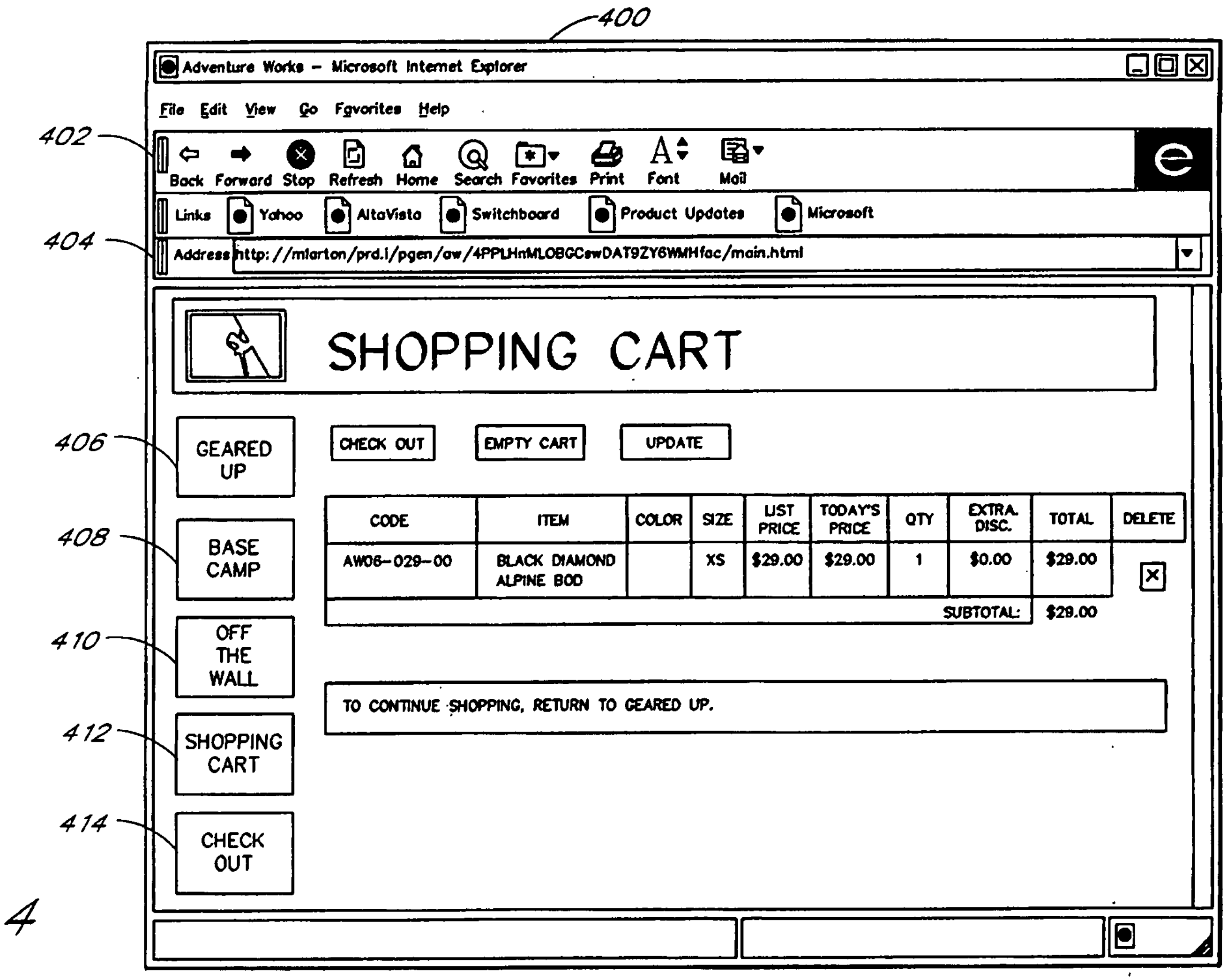

DATABASES
130
HTML STRUCTURES
128
MERCHANT STORE SERVER
212
STORE 1
STORE 2
STORE 3
STORE SERVER PROCESS 1
STORE SERVER PROCESS 2
STORE SERVER PROCESS 3
MERCHANT SYSTEM
106
106'
106'
204
ROUTER
INTERNET INFORMATION SERVER
202
GLOBAL CONFIGURATION CONTROLLER
206
104
108
108
108
102
102'
200
CONSUMER BROWSER
CONSUMER CLIENT
110
CONSUMER BROWSER
CONSUMER CLIENT
110'
MERCHANT BROWSER
MERCHANT CLIENT
210

FIG. 2

FIG. 3


STORE SERVER PROCESS
106
DATABASE
130
SHOPPER TABLE
PRODUCT TABLE
ORDER TABLE
DEPARTMENT TABLE
RECEIPT TABLE
PROMOTION TABLE
300
302
304
306
308
310
DATABASE MODULE
132
SHOPPER MANAGER
ORDER MANAGER
320
324
ORDER
124
ORDER BLACKBOARD
ITEM BLACKBOARD
350
352
ACCEPT STAGE
PAYMENT STAGE
INVENTORY STAGE
ORDER TOTAL STAGE
TAX STAGE
HANDLING STAGE
SHIPPING STAGE
ORDER PRICE ADJUST STAGE
ITEM PRICE ADJUST STAGE
ORDER CHECK STAGE
ORDER INITIALIZATION STAGE
SHOPPER INFORMATION STAGE
MERCHANT INFORMATION STAGE
PRODUCT INFORMATION STAGE
390
388
386
384
382
380
378
376
374
372
370
368
366
364
ORDER PIPELINE
ORDER PROCESSING MODULE
126
362
ORDER ENGINE
360
HTML TEMPLATES
128
DYNAMIC PAGE GENERATOR
120
ACTION MANAGER
122
order.additem action
order.clearform action
order.clearitems action
order.deleteitem action
order.deleteorder action
order.edititem action
order.editorder action
order.editquantities action
order.plan action
order.purchase action
330
332
334
336
338
340
342
344
346
348

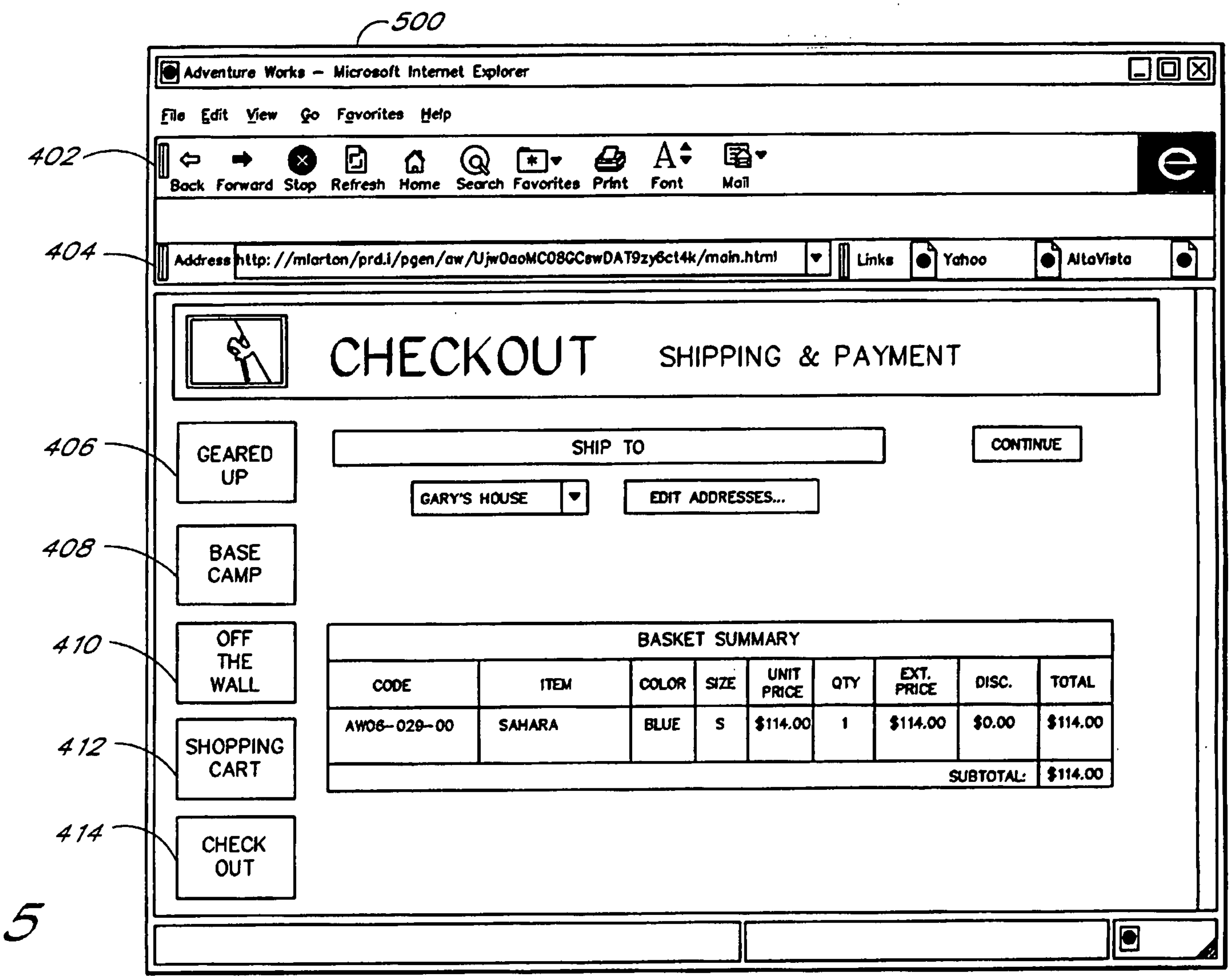

400
402
404
406
408
410
412
414
Adventure Works – Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail
Links Yahoo AltaVista Switchboard Product Updates Microsoft
Address http://mlarton/prd.l/pgen/aw/4PPLHnMLOBGCswDAT9ZY6WMHfac/main.html
GEARED UP
BASE CAMP
OFF THE WALL
SHOPPING CART
CHECK OUT
CHECK OUT
EMPTY CART
UPDATE

SHOPPING CART

CODE	ITEM	COLOR	SIZE	LIST PRICE	TODAY'S PRICE	QTY	EXTRA. DISC.	TOTAL	DELETE
AW06–029–00	BLACK DIAMOND ALPINE BOD		XS	$29.00	$29.00	1	$0.00	$29.00	☒
							SUBTOTAL:	$29.00	

TO CONTINUE SHOPPING, RETURN TO GEARED UP.

FIG. 4

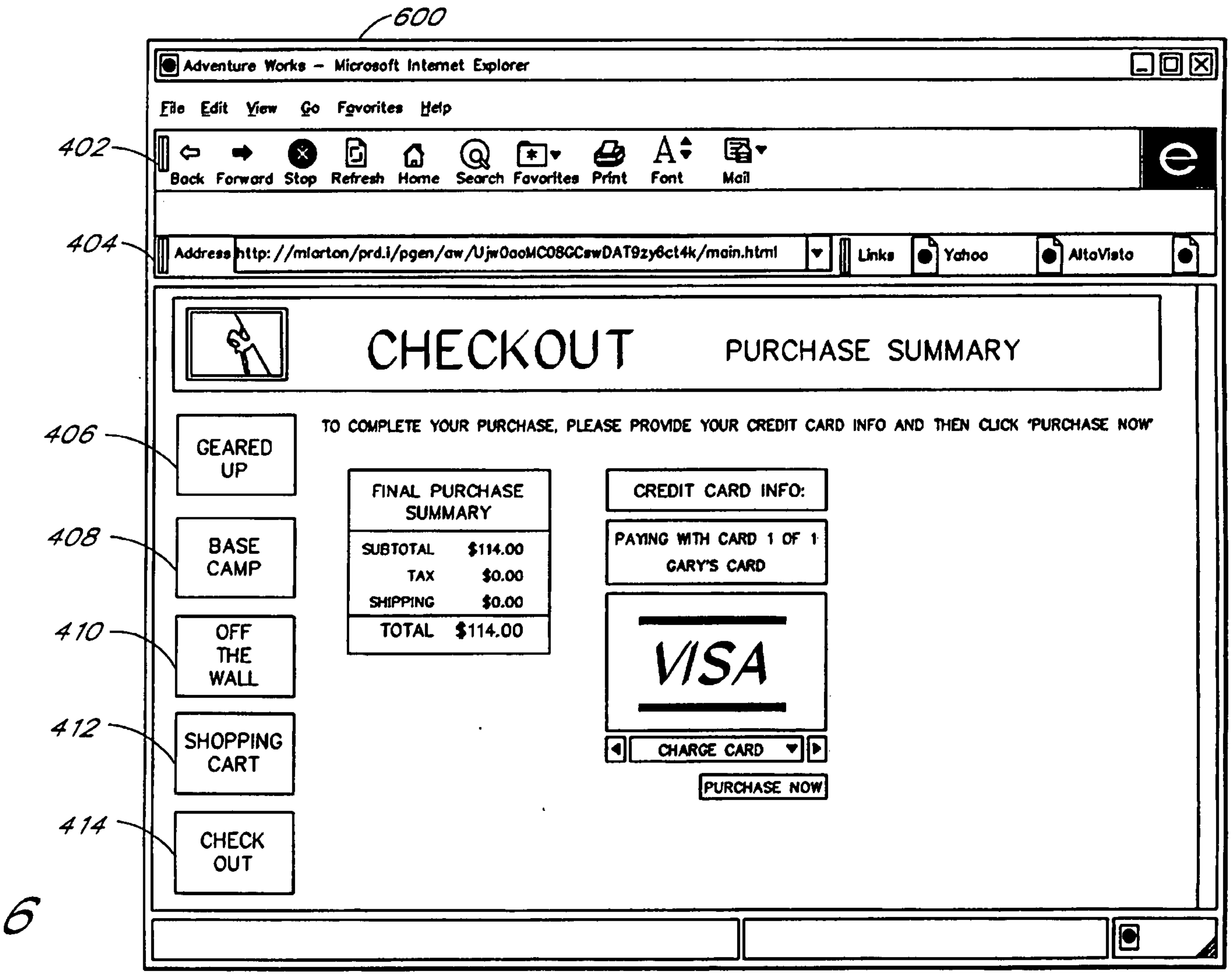

500
Adventure Works – Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail
Address http://mlarton/prd.i/pgen/aw/Ujw0aoMC08GCswDAT9zy6ct4k/main.html
Links Yahoo AltaVista
402
404
406
408
410
412
414
GEARED UP
BASE CAMP
OFF THE WALL
SHOPPING CART
CHECK OUT
SHIP TO
GARY'S HOUSE
EDIT ADDRESSES...
CONTINUE

CHECKOUT SHIPPING & PAYMENT

BASKET SUMMARY

CODE	ITEM	COLOR	SIZE	UNIT PRICE	QTY	EXT. PRICE	DISC.	TOTAL
AW06-029-00	SAHARA	BLUE	S	$114.00	1	$114.00	$0.00	$114.00
							SUBTOTAL:	$114.00

FIG. 5


600
Adventure Works – Microsoft Internet Explorer
File Edit View Go Favorites Help
402
Back Forward Stop Refresh Home Search Favorites Print Font Mail
404
Address http://mlorton/prd.i/pgen/aw/Ujw0aoMC08GCswDAT9zy6ct4k/main.html
Links Yahoo AltoVisto
CHECKOUT
PURCHASE SUMMARY
406
GEARED UP
408
BASE CAMP
410
OFF THE WALL
412
SHOPPING CART
414
CHECK OUT
CREDIT CARD INFO:
VISA
CHARGE CARD
PURCHASE NOW

TO COMPLETE YOUR PURCHASE, PLEASE PROVIDE YOUR CREDIT CARD INFO AND THEN CLICK 'PURCHASE NOW'

FINAL PURCHASE SUMMARY	
SUBTOTAL	$114.00
TAX	$0.00
SHIPPING	$0.00
TOTAL	$114.00

PAYING WITH CARD 1 OF 1
GARY'S CARD

FIG. 6

300

	SHOPPER IDENTIFIER	DATE CREATED	IS MEMBER	NAME	ADDRESS	CITY	STATE	ZIP	COUNTRY	• • •
1st SHOPPER										• • •
2nd SHOPPER										• • •
3rd SHOPPER										• • •
4th SHOPPER										• • •
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

FIG. 7

302 PRODUCT TABLES

800

	PRODUCT FAMILY IDENTIFIER	PRODUCT FAMILY NAME	PRODUCT FAMILY DESCRIPTION	DEPARTMENT IDENTIFIER	SIZE TYPE	DATE INTRODUCED	LIST PRICE	SALE PRICE	SALE START	SALE END	IMAGE FILE NAME	• • •
1st PRODUCT FAMILY												• • •
2nd PRODUCT FAMILY												• • •
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

PRODUCT FAMILY TABLE

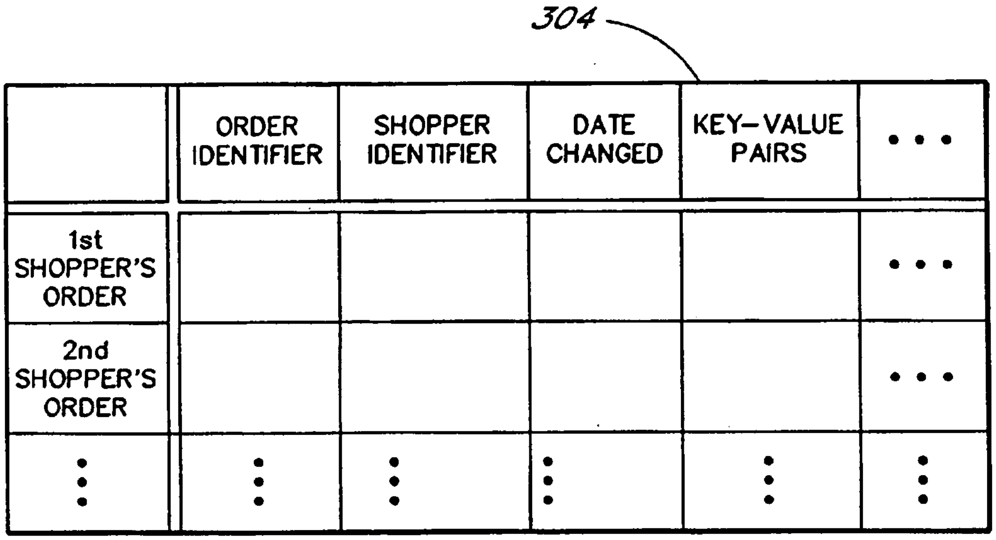
FIG. 8A

802

	PRODUCT FAMILY IDENTIFIER	SKU	COLOR VALUE	SIZE VALUE	• • •
1st PRODUCT VARIANT					• • •
2nd PRODUCT VARIANT					• • •
⋮	⋮	⋮	⋮	⋮	⋮

PRODUCT VARIANT TABLE

FIG. 8B



	ORDER IDENTIFIER	SHOPPER IDENTIFIER	DATE CHANGED	KEY-VALUE PAIRS	• • •
1st SHOPPER'S ORDER					• • •
2nd SHOPPER'S ORDER					• • •
⋮	⋮	⋮	⋮	⋮	⋮

ORDER TABLE

FIG. 9

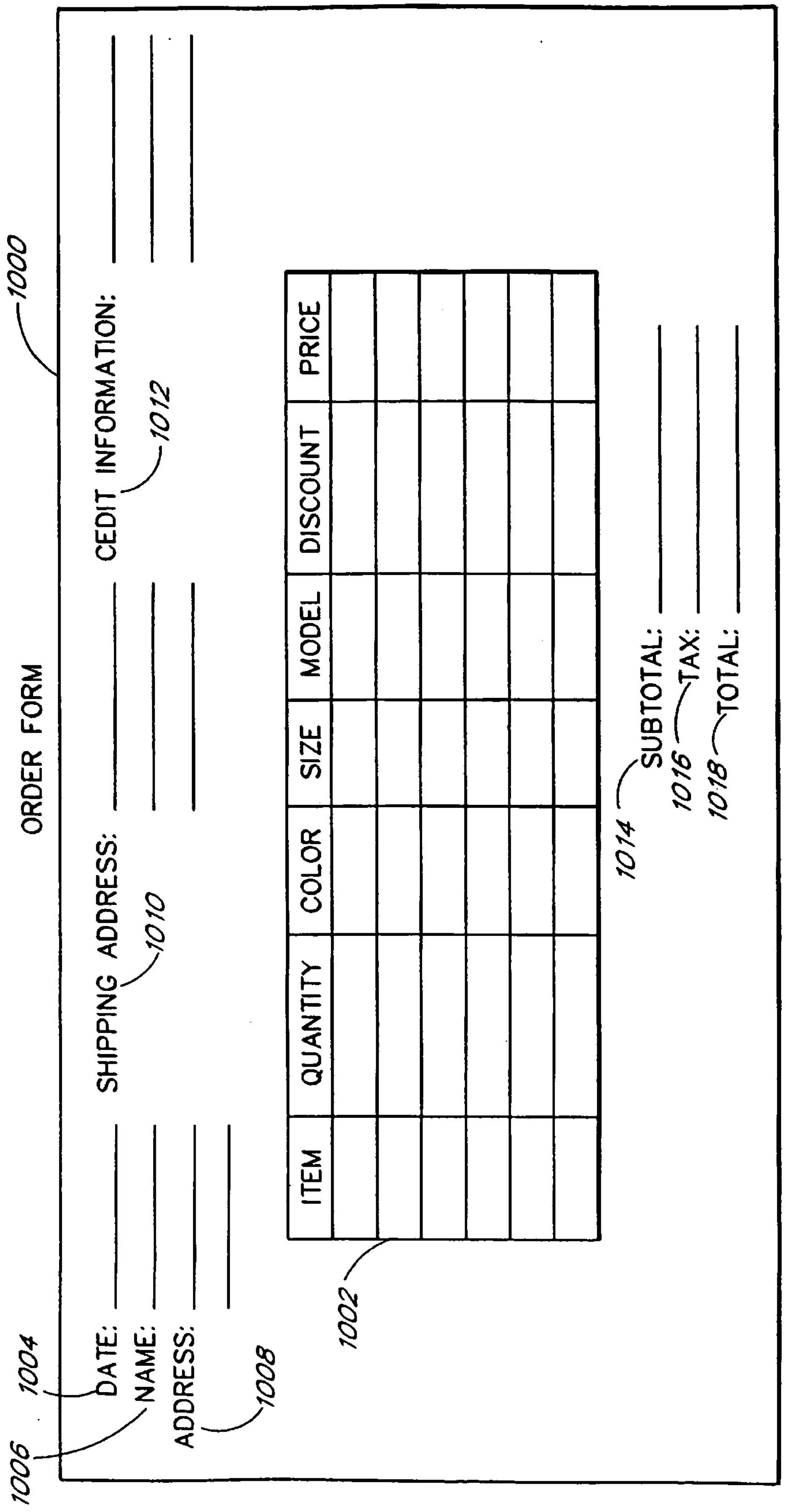


FIG. 10
(PRIOR ART)

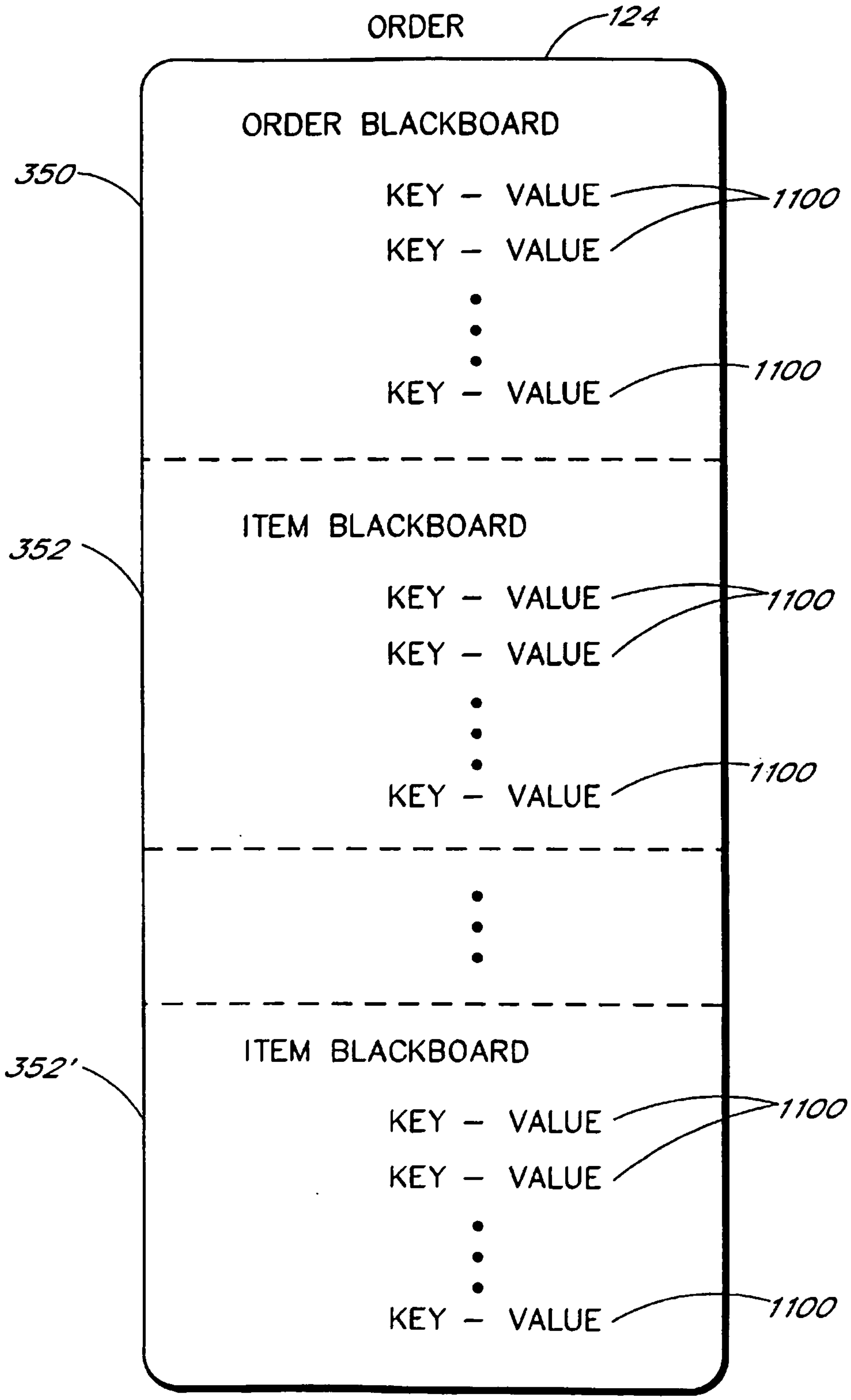

ORDER
124
ORDER BLACKBOARD
350
KEY – VALUE
KEY – VALUE
KEY – VALUE
1100
1100
ITEM BLACKBOARD
352
KEY – VALUE
KEY – VALUE
KEY – VALUE
1100
1100
ITEM BLACKBOARD
352'
KEY – VALUE
KEY – VALUE
KEY – VALUE
1100
1100

FIG. 11

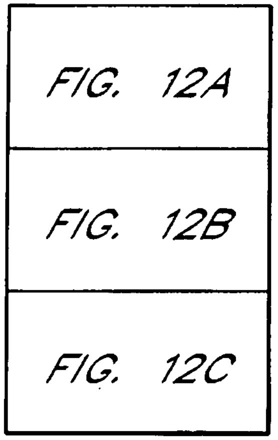


FIG. 12

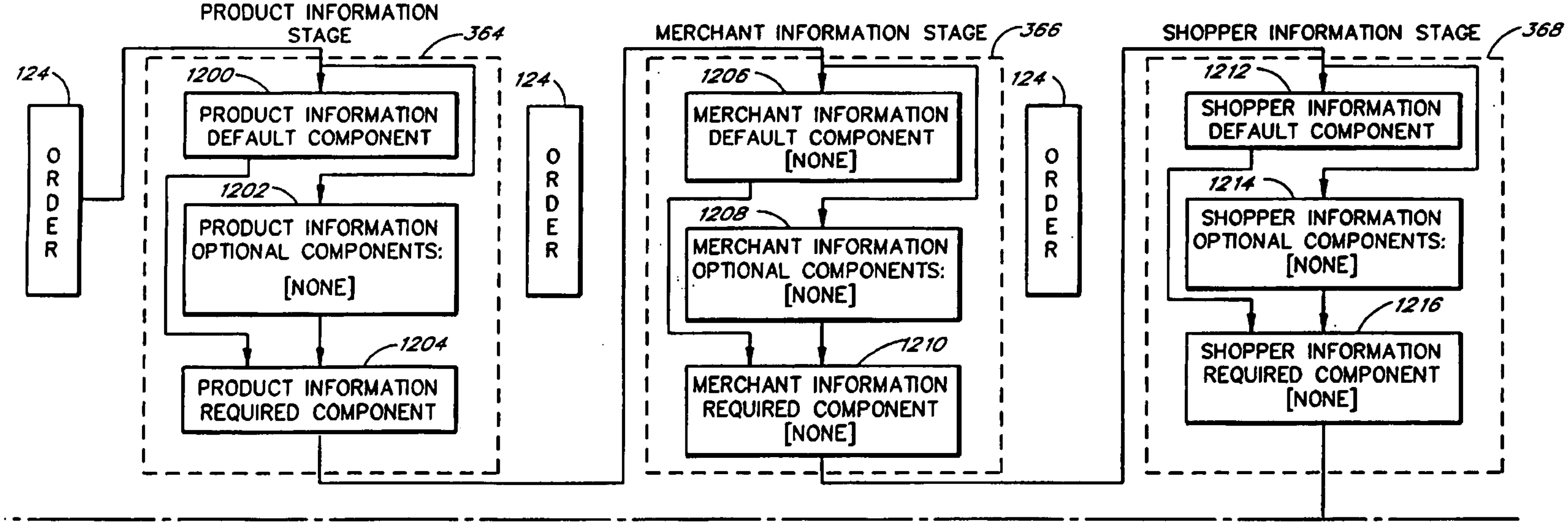

124
ORDER
PRODUCT INFORMATION STAGE
364
1200
PRODUCT INFORMATION DEFAULT COMPONENT
1202
PRODUCT INFORMATION OPTIONAL COMPONENTS: [NONE]
1204
PRODUCT INFORMATION REQUIRED COMPONENT
124
ORDER
MERCHANT INFORMATION STAGE
366
1206
MERCHANT INFORMATION DEFAULT COMPONENT [NONE]
1208
MERCHANT INFORMATION OPTIONAL COMPONENTS: [NONE]
1210
MERCHANT INFORMATION REQUIRED COMPONENT [NONE]
124
ORDER
SHOPPER INFORMATION STAGE
368
1212
SHOPPER INFORMATION DEFAULT COMPONENT
1214
SHOPPER INFORMATION OPTIONAL COMPONENTS: [NONE]
1216
SHOPPER INFORMATION REQUIRED COMPONENT [NONE]

FIG. 12A

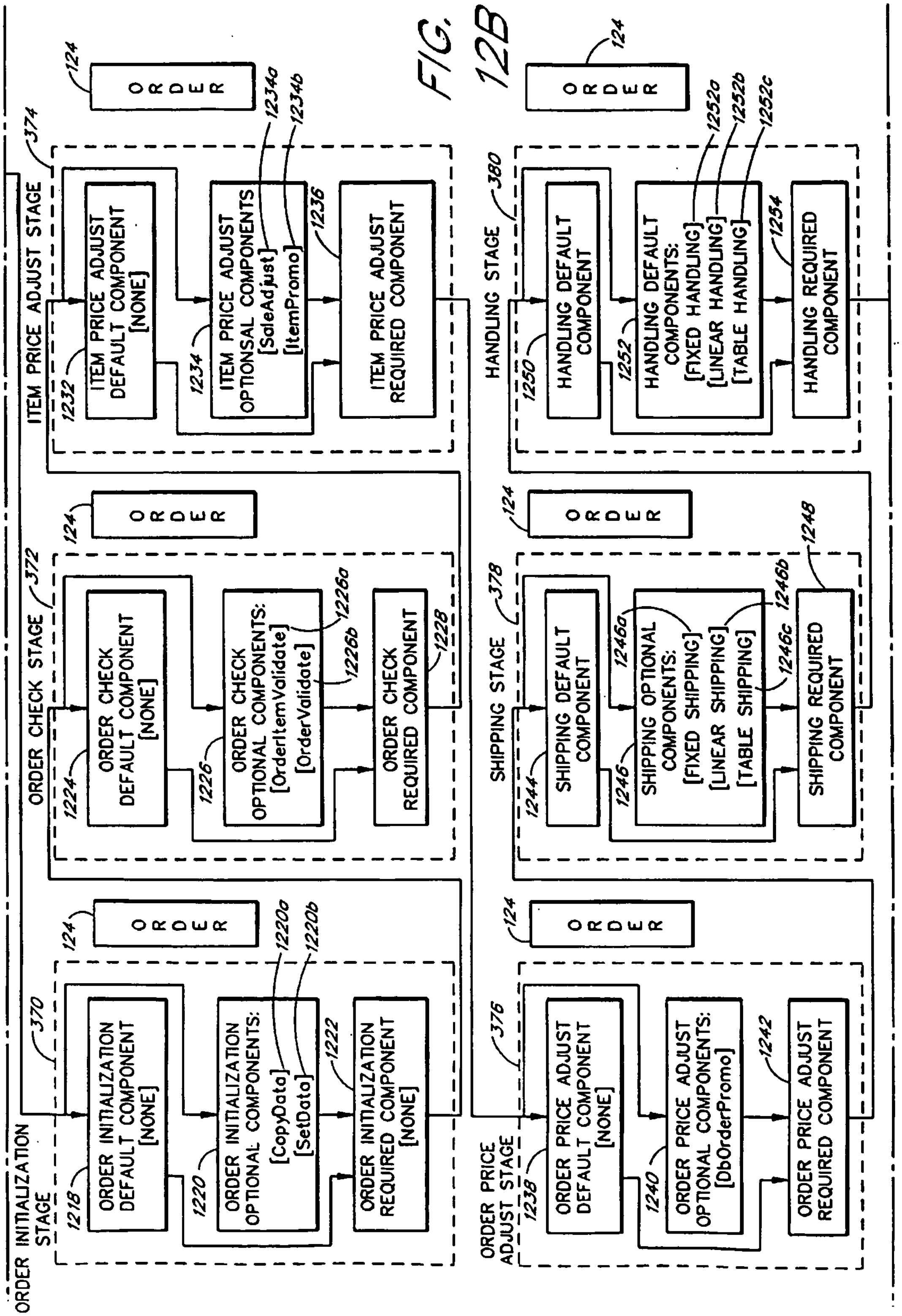


FIG. 12B

ORDER INITIALIZATION STAGE
370
1218
ORDER INITIALIZATION DEFAULT COMPONENT [NONE]
1220
ORDER INITIALIZATION OPTIONAL COMPONENTS: [CopyData] [SetData]
1220a
1220b
1222
ORDER INITIALIZATION REQUIRED COMPONENT [NONE]
124
ORDER
ORDER CHECK STAGE
372
1224
ORDER CHECK DEFAULT COMPONENT [NONE]
1226
ORDER CHECK OPTIONAL COMPONENTS: [OrderItemValidate] [OrderValidate]
1226a
1226b
1228
ORDER CHECK REQUIRED COMPONENT
124
ORDER
ITEM PRICE ADJUST STAGE
374
1232
ITEM PRICE ADJUST DEFAULT COMPONENT [NONE]
1234
ITEM PRICE ADJUST OPTIONSAL COMPONENTS [SaleAdjust] [ItemPromo]
1234a
1234b
1236
ITEM PRICE ADJUST REQUIRED COMPONENT
124
ORDER
ORDER PRICE ADJUST STAGE
376
1238
ORDER PRICE ADJUST DEFAULT COMPONENT [NONE]
1240
ORDER PRICE ADJUST OPTIONAL COMPONENTS: [DbOrderPromo]
1242
ORDER PRICE ADJUST REQUIRED COMPONENT
124
ORDER
SHIPPING STAGE
378
1244
SHIPPING DEFAULT COMPONENT
1246
SHIPPING OPTIONAL COMPONENTS: [FIXED SHIPPING] [LINEAR SHIPPING] [TABLE SHIPPING]
1246a
1246b
1246c
1248
SHIPPING REQUIRED COMPONENT
124
ORDER
HANDLING STAGE
380
1250
HANDLING DEFAULT COMPONENT
1252
HANDLING DEFAULT COMPONENTS: [FIXED HANDLING] [LINEAR HANDLING] [TABLE HANDLING]
1252a
1252b
1252c
1254
HANDLING REQUIRED COMPONENT
124
ORDER

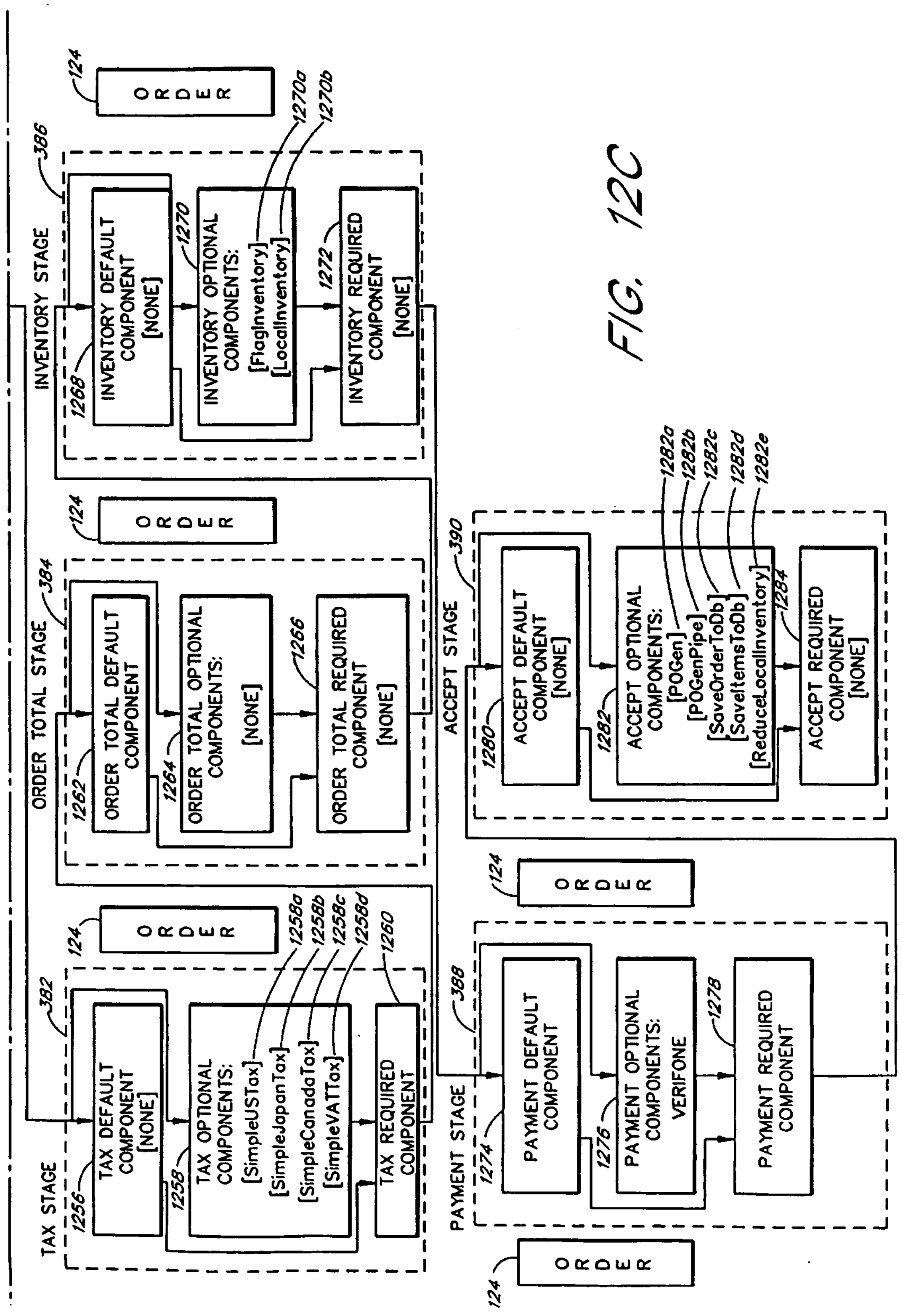

TAX STAGE
382
1256
TAX DEFAULT COMPONENT [NONE]
1258
TAX OPTIONAL COMPONENTS: [SimpleUSTax] [SimpleJapanTax] [SimpleCanadaTax] [SimpleVATTax]
1258a
1258b
1258c
1258d
1260
TAX REQUIRED COMPONENT
124
ORDER
ORDER TOTAL STAGE
384
1262
ORDER TOTAL DEFAULT COMPONENT
1264
ORDER TOTAL OPTIONAL COMPONENTS: [NONE]
1266
ORDER TOTAL REQUIRED COMPONENT [NONE]
124
ORDER
INVENTORY STAGE
386
1268
INVENTORY DEFAULT COMPONENT [NONE]
1270
INVENTORY OPTIONAL COMPONENTS: [FlagInventory] [LocalInventory]
1270a
1270b
1272
INVENTORY REQUIRED COMPONENT [NONE]
124
ORDER
PAYMENT STAGE
388
1274
PAYMENT DEFAULT COMPONENT
1276
PAYMENT OPTIONAL COMPONENTS: VERIFONE
1278
PAYMENT REQUIRED COMPONENT
124
ORDER
ACCEPT STAGE
390
1280
ACCEPT DEFAULT COMPONENT [NONE]
1282
ACCEPT OPTIONAL COMPONENTS: [POGen] [POGenPipe] [SaveOrderToDb] [SaveItemsToDb] [ReduceLocalInventory]
1282a
1282b
1282c
1282d
1282e
1284
ACCEPT REQUIRED COMPONENT [NONE]
124
ORDER

FIG. 12C

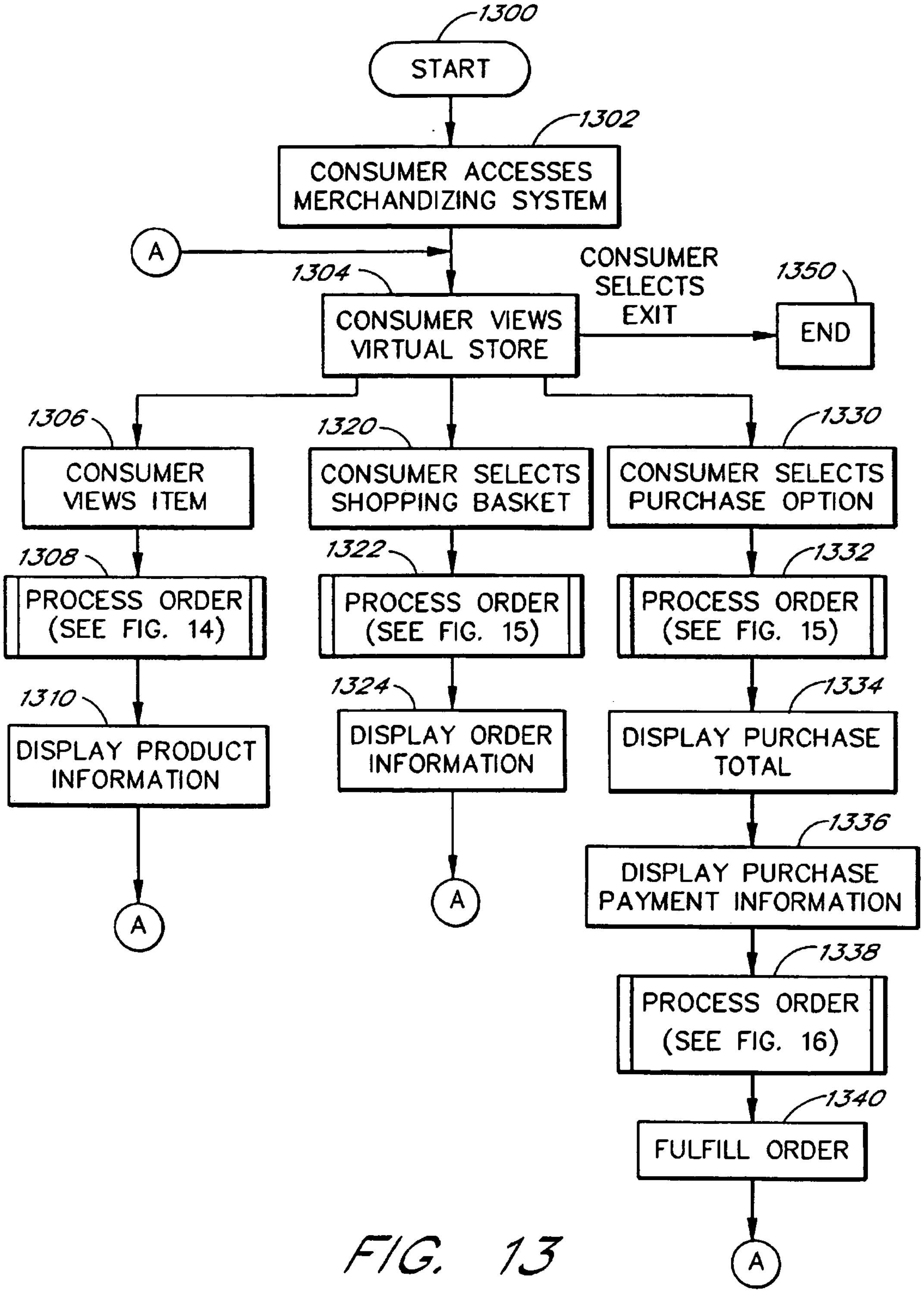

1300
START
1302
CONSUMER ACCESSES MERCHANDIZING SYSTEM
A
1304
CONSUMER VIEWS VIRTUAL STORE
CONSUMER SELECTS EXIT
1350
END
1306
CONSUMER VIEWS ITEM
1308
PROCESS ORDER (SEE FIG. 14)
1310
DISPLAY PRODUCT INFORMATION
A
1320
CONSUMER SELECTS SHOPPING BASKET
1322
PROCESS ORDER (SEE FIG. 15)
1324
DISPLAY ORDER INFORMATION
A
1330
CONSUMER SELECTS PURCHASE OPTION
1332
PROCESS ORDER (SEE FIG. 15)
1334
DISPLAY PURCHASE TOTAL
1336
DISPLAY PURCHASE PAYMENT INFORMATION
1338
PROCESS ORDER (SEE FIG. 16)
1340
FULFILL ORDER
A

FIG. 13

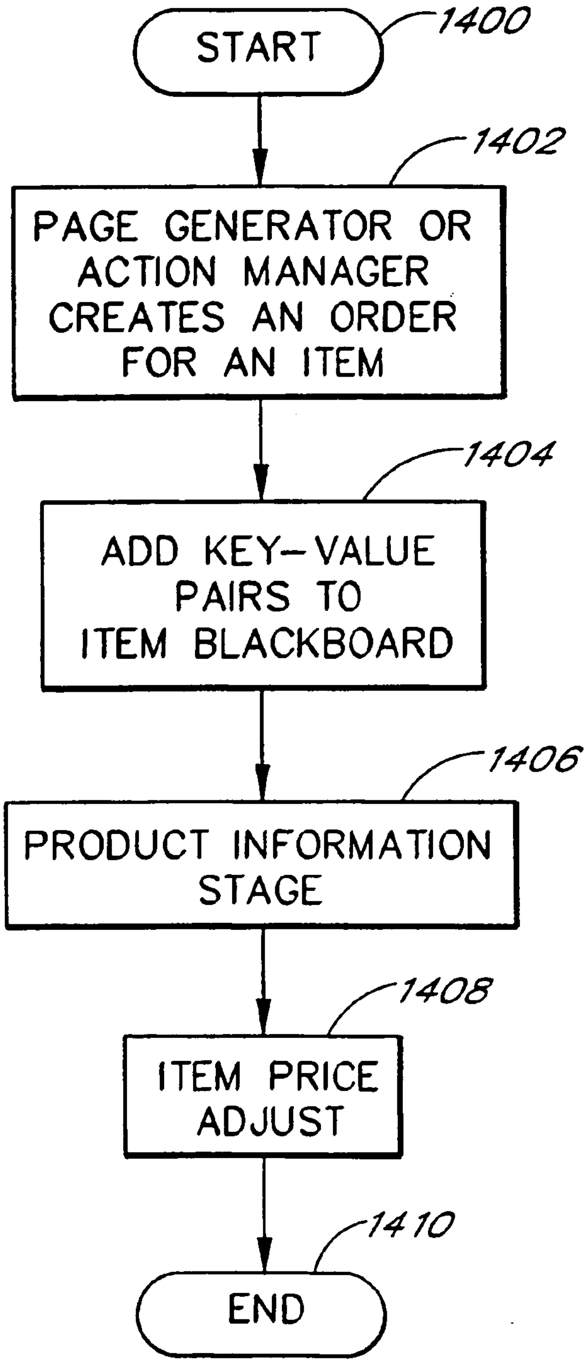

START
1400
1402
PAGE GENERATOR OR ACTION MANAGER CREATES AN ORDER FOR AN ITEM
1404
ADD KEY-VALUE PAIRS TO ITEM BLACKBOARD
1406
PRODUCT INFORMATION STAGE
1408
ITEM PRICE ADJUST
1410
END

FIG. 14

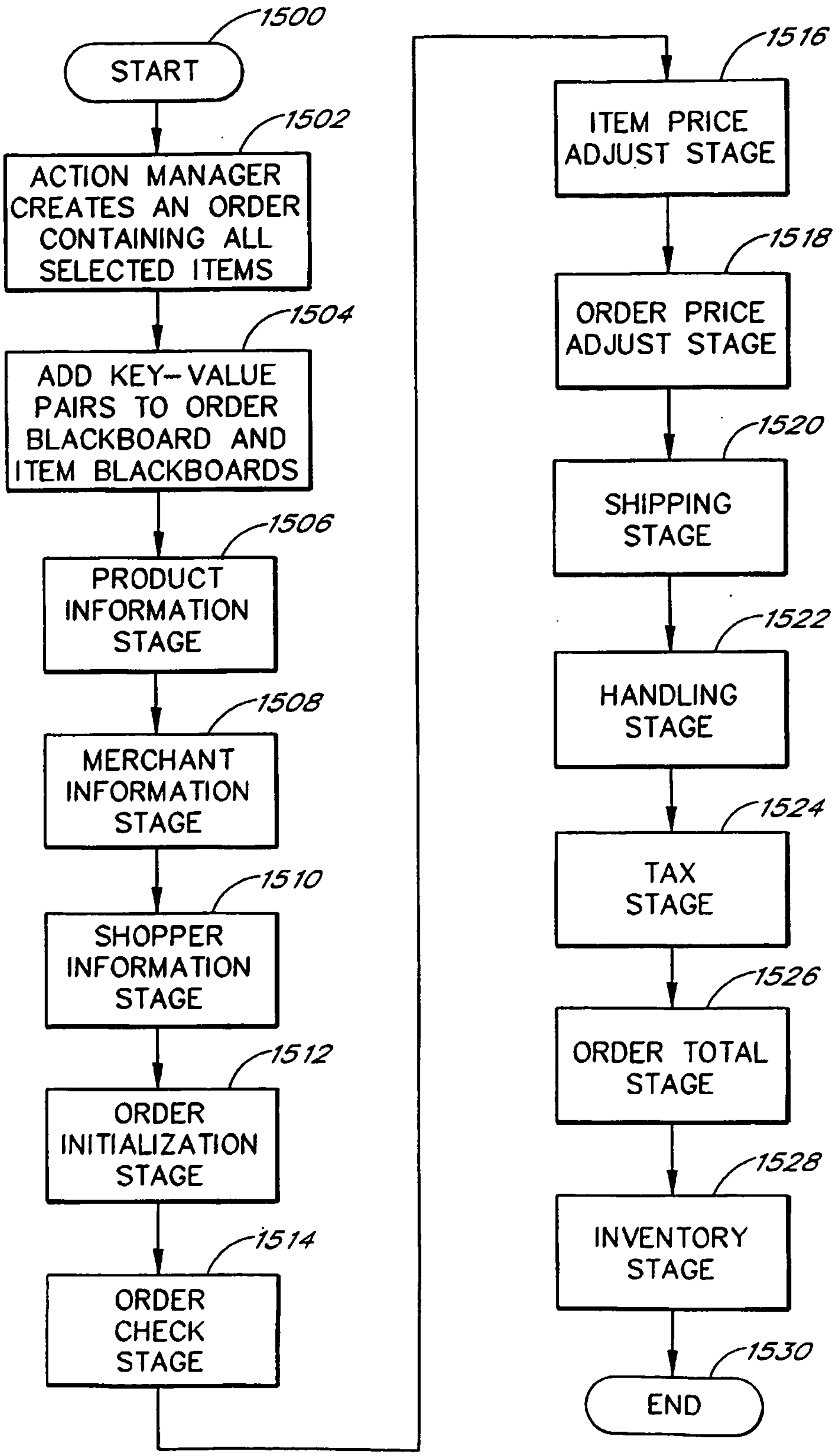

1500
START
1502
ACTION MANAGER CREATES AN ORDER CONTAINING ALL SELECTED ITEMS
1504
ADD KEY-VALUE PAIRS TO ORDER BLACKBOARD AND ITEM BLACKBOARDS
1506
PRODUCT INFORMATION STAGE
1508
MERCHANT INFORMATION STAGE
1510
SHOPPER INFORMATION STAGE
1512
ORDER INITIALIZATION STAGE
1514
ORDER CHECK STAGE
1516
ITEM PRICE ADJUST STAGE
1518
ORDER PRICE ADJUST STAGE
1520
SHIPPING STAGE
1522
HANDLING STAGE
1524
TAX STAGE
1526
ORDER TOTAL STAGE
1528
INVENTORY STAGE
1530
END

FIG. 15

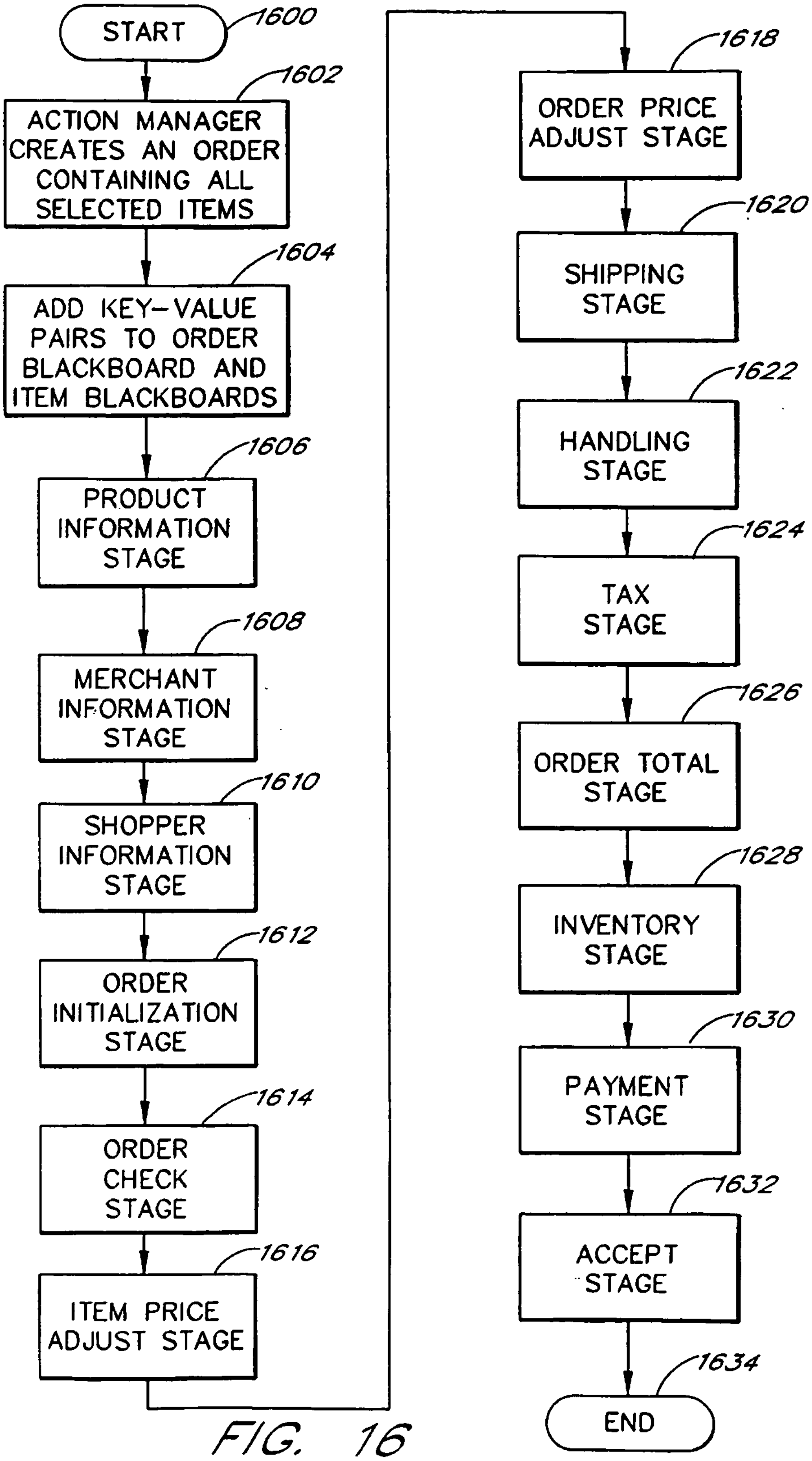

START
1600
ACTION MANAGER CREATES AN ORDER CONTAINING ALL SELECTED ITEMS
1602
ADD KEY-VALUE PAIRS TO ORDER BLACKBOARD AND ITEM BLACKBOARDS
1604
PRODUCT INFORMATION STAGE
1606
MERCHANT INFORMATION STAGE
1608
SHOPPER INFORMATION STAGE
1610
ORDER INITIALIZATION STAGE
1612
ORDER CHECK STAGE
1614
ITEM PRICE ADJUST STAGE
1616
ORDER PRICE ADJUST STAGE
1618
SHIPPING STAGE
1620
HANDLING STAGE
1622
TAX STAGE
1624
ORDER TOTAL STAGE
1626
INVENTORY STAGE
1628
PAYMENT STAGE
1630
ACCEPT STAGE
1632
END
1634

FIG. 16

SYSTEM AND METHOD FOR PROCESSING ELECTRONIC ORDER FORMS

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to computer network communication systems and, more particularly, to electronic merchandising systems which allow consumers to purchase goods and services over a distributed network.

2. Background

Electronic merchandising systems currently exist which allow users to purchase goods and services from a variety of different merchants over a distributed computer network such as the Internet. With systems of this type, the merchants typically establish a virtual store which end users can interactively view with a personal computer which is connected to the network. The end users or consumers can then purchase desired items offered for sale.

In World Wide Web ("Web") based implementations, the virtual stores are in the form of hypertext documents which are hosted by the Web sites of the respective merchants. Typically, a Web site is an Internet-connected computer or computer system which runs server software for serving information using the standard protocols of the World Wide Web. In other implementations, the merchants' hypertext documents may, for example, be hosted by a centralized computer of an on-line services network, such as the Microsoft Network (MSN), or by an Internet site which is accessed using proprietary applications software.

Conventional hypertext documents contain pictures, textual descriptions, and pricing information with respect to the products and/or services offered by the respective merchants. In addition, the hypertext documents include electronic order forms which allow consumers to purchase the goods and services offered by the merchants. The hypertext documents are typically accessed using a standard Web browser application which runs on the consumer's computer.

For example, a consumer may direct his Web browser to access a merchant's hypertext documents. Upon viewing a desired good or service, the consumer fills out an electronic order form which specifies the name of the consumer, a shipping address, billing information, the desired good or service, etc. The consumer's Web browser then transmits the electronic order form to the Merchant's Website. Upon receiving the electronic order form, the Merchant Website processes the electronic order form to complete the sales transaction.

Prior art electronic merchandising systems, however, typically use electronic order forms comprising rigid records with fixed attributes. Consequently, conventional electronic order forms are not easily adaptable to the rapidly changing electronic sales environment. Furthermore, changes in the electronic order form often require corresponding changes in the electronic purchasing system.

In addition, conventional electronic merchandising systems process the electronic order forms with modules which typically require a significant level of inter-module communication. The inter-module communication usually requires significant order processing resources. In addition, the complex interrelationships which exist between the modules make any changes to the purchasing process a time consuming and risky task.

SUMMARY OF THE INVENTION

The present invention provides a method and system for processing electronic sales transactions. In a preferred embodiment, the electronic merchandising system allows merchants to create electronic orders which are easily adaptable for different sales situations. The preferred electronic order comprises flexible blackboards which allow merchants to add sales information with what are called key-value pairs. The sales transaction information stored in the key-value pairs may include, by a way of example, special shipping information, unique billing information, gift wrap information, monogram information, etc.

Unlike other order forms which rely on specific database structures or other structured formats, the order of the present invention contains as many key-value pairs as are necessary to define a sales transaction. This allows merchants to easily customize the electronic merchandising system for diverse sales transactions. Advantageously, merchants can add new key-value pairs or delete existing key-value pairs to modify the electronic merchandizing system.

For example, a conventional order form typically contains a list of purchased items and other order information. The order information, also referred to as the order properties, the name of the shopper, the shopper's address, the shipping address, the order subtotal, the taxes and the order total amount. The list of items typically contains an entry for each item and item information such as the quantity of items, the color, size and model of items, the item discount, item price, etc.

Such order forms will vary from merchant to merchant. For example, an international merchant may require complex tax information, a merchant which provides gift wrapping will need gift wrapping information, a merchant which provides monogram services will need monogram information, etc. In addition, in the electronic merchandising business, new services and billing methods are being added at a rapid pace. For example, if a merchant accepts digital cash, the merchant will have to modify the billing information to accept digital cash information.

Unfortunately, conventional electronic merchandising systems typically represent the order information and item information in predefined formats such as a database format with specific fields. If new payment methods are needed or if a merchant has unique needs, the merchant must revise the specified order form format. Unfortunately, revising the order form format can result in significant revisions to the electronic merchandising software.

One approach in conventional electronic merchandising systems is to predefine every data element which may be needed in any sales transactions. For example, some companies have attempted to define every data element which is needed for shipping information. Unfortunately, even if every possible data element needed to represent current sales transactions could be predefined, new sales transactions would arise which require new revisions to the electronic merchandising software.

Rather than utilizing a predefined organization of data elements, the present invention utilizes an order with multiple key-value pairs which are not organized with a predefined format. In the preferred embodiment, the order is an object which contains at least one order blackboard and one or more item blackboards. Preferably, each blackboard contains a set of key-value pairs. Each key-value pair, in turn, contains a value and a key which identifies the value. In the order blackboard, the key-value pairs contain order properties such as the consumer's name, the consumer's address, the desired shipping address, the billing information, the order subtotal, the taxes, the order total, etc.

The key-value pairs in the item blackboards contain information about each item. Preferably, an item blackboard exists for each item. Furthermore, the key-value pairs in one item blackboard can differ from the key-value pairs in another item blackboard. For example, the key-value pairs in one of the item blackboards contain, but are not limited to, 1) the information which defines a particular item such as the item stock keeping unit (sku) and item quantity, 2) the information sent from the consumer to the merchant which defines additional services associated with the item (i.e., a monogram service or a gift wrapping service) and 3) information known to the merchant and kept on the item blackboard for reference such as an item description, an item size, an item price, etc.

In the preferred embodiment, an order processing module processes the order. The order processing module contains an order engine and multiple components called the order pipeline. The order engine determines which components in the order pipeline process the order. Each component in the order pipeline reads from or writes to its assigned key-value pairs. Upon receiving an order form, a component searches for its assigned key-value pairs and adds its own key-value pairs necessary to process the order.

Thus, each component only modifies its assigned key-value pairs. This allows a merchant to add new key-value pairs without having to also modify the software instructions in the existing order processing components. For example, assume that a merchant sells shirts. Furthermore, assume that the merchant desires to provide a monogram service which adds a consumer's initials to the purchased shirts. As discussed in more detail below, the merchant defines within the merchant software, a monogram key-value pair which comprises a monogram key word and a corresponding value which stores the consumer's initials. Furthermore, the merchant adds a customized monogram component to the order processing module.

In this example, when the consumer purchases a shirt with the monogram service option, the consumer enters his initials. The electronic merchandising system then creates an order with the key-value pairs necessary to complete the sales transaction. In this example, the electronic merchandising system also adds the monogram key-value pair to the order.

Each component in the order processing system processes its assigned key-value pairs. When the monogram component receives an electronic order, the monogram component searches the electronic order for the monogram key-value pair and performs the necessary steps to ensure that the appropriate amount is billed for the monogram service.

Thus, the preferred embodiment of the present invention allows the merchants to customize the electronic merchandising system for different sales situations with a minimal amount of programming effort. Rather than having to alter existing order processing components for different sales transactions, a merchant can modify the existing component or replace an existing component with a new component. Accordingly, the programing effort required to modify the existing order processing module is greatly reduced.

For example, suppose a merchant has an order processing system which computes the price of an order. Furthermore, assume that the merchant desires to provide a promotional discount. With the preferred embodiment of the present invention, the merchant simply adds 1) a promotion key-value pair to the electronic order and 2) a promotion component to the order processing system. In this example, the promotion key-value pair identifies that a promotional discount may apply to the electronic order. When the promotion component receives the electronic order, the promotion component performs the necessary calculations to discount the price of the electronic order based on the contents of the discount key-value pair.

In another aspect of the present invention, the components in the order pipeline use the order as their primary means of intercommunication. If a first component needs to pass information to a second component, the first component stores the appropriate key-value pair in the order. Thus, the components communicate with each other by storing information in the order. This reduces the system requirements needed to support inter-component communication.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other aspects, advantages, and novel features of the invention will become apparent upon reading the following detailed description and upon reference to accompanying drawings in which:

FIG. 1 is a high level block diagram illustrating the preferred electronic merchandising system.

FIG. 2 is a block diagram illustrating the architecture of the preferred electronic merchandising system.

FIG. 3 is a block diagram illustrating the preferred modules in the store server process.

FIG. 4 is a diagram illustrating one embodiment of a graphical user interface for a shopping cart in the preferred embodiment of the present invention.

FIG. 5 is a diagram illustrating one embodiment of a graphical user interface for an order total in the preferred embodiment of the present invention.

FIG. 6 is a diagram illustrating one embodiment of a graphical user interface for payment instructions in the preferred embodiment of the present invention.

FIG. 7 illustrates one embodiment of a shopper table.

FIG. 8A and 8B illustrates one embodiment of the product tables.

FIG. 9 is a block diagram of one embodiment of a shopper table.

FIG. 10 is a block diagram of a prior art order form.

FIG. 11 illustrates the format of an order in the preferred embodiment of the present invention.

FIGS. 12A–12C illustrate the components in the preferred order pipeline.

FIG. 13 illustrates a flow chart of a preferred shopping process.

FIG. 14 illustrates a flow chart of a preferred product display process.

FIG. 15 illustrates a flow chart of a preferred order total process.

FIG. 16 illustrates a flow chart of a preferred order completion process.

In the drawings, the first digit of any three-digit number indicates the number of the figure in which the element first appears. For example, an element with the reference number 402 first appears in FIG. 4. In addition, like reference numerals are used throughout the drawings to indicate correspondence between elements.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention provides a method and system for processing electronic sales transactions. In a preferred embodiment, an electronic merchandising system allows merchants to create electronic orders which are easily adaptable for different sales situations. The preferred electronic order comprises flexible blackboards which allow merchants to add key-value pairs. The sales transaction information stored in the key-value pairs may include, by a way of example, special shipping information, unique billing information, gift wrap information, monogram information, etc.

Although the preferred embodiment is described herein with reference to a preferred computer networking system, the invention is not so limited, and can be used in a variety of other contexts in which it is desirable to adapt electronic order for different situations. To facilitate a complete understanding of the invention, the remainder of the detailed description is organized into the following sections and subsections:

I. Glossary Of Terms And Acronyms

Action. An action performs various functions in the electronic merchandising system such as, by way of example, adding an item to an order form, beginning a purchase process, or inserting or deleting data from a database.

Cluster. A cluster refers to two or more merchandising computers which are configured to execute the merchant system.

Client-Server. A model of interaction in a computer system in which one program sends a request to another program. The requesting program is called the "client," and the program which responds to the client is called the "server." In the context of the World Wide Web, the preferred client is typically called a "Web browser" which runs on a user's computer. The preferred server program which responds to a Web browser's requests is commonly referred to as a "Web server."

Guid (Globally Unique Identifier). A globally unique identifier is a unique 128-bit value which is typically created with a define GUID routine. GUID definition routines are well known in the art and further described in the *OLE* 2 *Programmer's Reference Volumes I and II,* Microsoft Press, 1993 and Brockschmidt, *Inside OLE* 2, Microsoft Press, 1994.

Internet. A collection of interconnected (public and private) networks which are linked together by a set of standard protocols to form a distributed network. While this term is intended to refer to what is now commonly known as the Internet, it is also intended to encompass intranet implementations, and variations which may be made in the future, including changes and additions to existing networks and network protocols.

HyperText Markup Language (HTML). A standard coding convention and set of codes for attaching presentation and linking attributes to informational content within documents. (HTML 2.0 is currently the primary standard used for generating Web documents.) During a document authoring stage, the HTML codes (referred to as "tags") are embedded within the informational content of the document. When the Web document (or "HTML document") is subsequently transferred from a Web server to a Web browser, the codes are interpreted by the Web browser and used to parse and display the document. In addition to specifying how the Web browser is to display the document, HTML tags can be used to create links to other websites and other Web documents (commonly referred to as "hyperlinks"). Such hyperlinks typically contain a Universal Resource Locator (URL) which identifies an HTML document in a Website. For more information on HTML, see Ian S. Gram, *The HTML Source Book,* John Wiley and Sons, Inc., 1995.

HyperText Transport Protocol (HTTP). The standard World Wide Web client-server protocol used for the exchange of information (such as HTML documents, and client requests for such documents) between a Web browser and a Web server. HTTP includes a number of different message types which can be sent from the client to the server to request different server actions. For example, a "GET" message, which has the format GET <URL>, causes the server to return the document or file specified by the Universal Resource Locator (URL).

ISAPI (Internet Server Application Program Interface). In the preferred embodiment, ISAPI is Microsoft's interface for allowing a Web server (or other information server) to launch and interact with external programs in response to requests from clients. ISAPI programs are in the form of dynamic link libraries (DLLs) which run in the same process space as the Web server. Documentation on ISAPI is available from Microsoft Corporation as part of the Microsoft Internet Information Server Software Development kit.

Named Pipes. A named pipe is an interprocess communication method used to transfer data between separate processes, usually on separate computers. For more information on pipes see Abraham Silberschatz and Peter B. Galvin, *Operating System Concepts,* Addison-Wesley Publishing Company, 1994.

Persistent Client State Cookies (Cookie). A file which stores information on the client for use by the server. In the preferred embodiment, the web browser stores consumer information in the cookie so that the consumer does not need to retype the consumer information each time the consumer accesses the electronic merchandising system. The specification for Cookies can be found at http://www.netscape.com/newsref/std/cookie_spec.html.

Registry. The registry is a repository for the information about the computer hardware and the software installed in a computer. In the preferred embodiment, the registry for the electronic merchandizing system is the Windows NT registry which is part of the Windows NT operating system available from Microsoft Corporation. The Windows NT registry and the methods for modifying the Windows NT registry are well known in the art, and are described in *The Win 32 Programmer's Reference Vol. II,* Microsoft Press, 1993. The registry can however, include a wide variety of registry mechanisms such as a database, a flat file and various user interfaces.

SKU. The sku acronym means the stock keeping unit. The stock keeping unit is a value which uniquely identifies each item in a merchant's inventory of items.

Transmission Control ProtocoI/Internet Protocol (TCP/IP). A standard Internet protocol (or set of protocols) which specifies how two computers exchange data over the Internet. TCP/IP handles issues such as packetization, packet addressing, handshaking and error correction. For more information on TCP/IP, see Volumes I, II and III of Corner and Stevens, *Internetworking with TCP/IP,* Prentice Hall, Inc., ISBN 0-13468505-9 (Vol. I), 0-13-125527-4 (Vol. II), and 0-13-474222-2 (Vol. III).

Uniform Resource Locator (URL). A unique address which fully specifies the location of a file or other resource on the Internet. The general format of a URL is protocol://machineaddress/path/filename.

World Wide Web ("Web"). Used herein to refer generally to both 1) a distributed collection of interlinked, user-viewable hypertext documents (commonly referred to as "Web documents" or "electronic pages" or "home pages") that are accessible via the Internet, and 2) the client and server software components which provide user access to such documents using standardized Internet protocols. Currently, the standard protocol for allowing applications to locate and acquire Web documents is the HyperText Transfer Protocol (HTTP), and the electronic pages are encoded using the HyperText Markup Language (HTML). However, the terms "Web" and "World Wide Web" are intended to encompass future markup languages and transport protocols which may be used in place of, or in addition to, the HyperText Markup Language and the HyperText Transfer Protocol.

II. Overview Of The Preferred Electronic Merchandising System

This section provides an overview of the preferred electronic merchandising system **100** as illustrated in FIG. 1. The preferred electronic merchandising system **100** is a client/server system which allows merchants to provide a virtual store which processes sales transactions. The electronic merchandising system **100** includes a consumer client **102** (the client) and a merchant system **104** which contains at least one store server process **106** (the server). The consumer client **102** and store server process **106** are in communication with each other by use of a communications medium **108**.

The consumer client **102** contains a consumer browser **110**. The consumer browser **110** communicates with the store server process **106** and displays the web documents created by the store server process **106**. Each store server process **106** provides a server architecture that supports the presentation and administration of a virtual store. Preferably, the store server process **106** comprises a number of components including a dynamic page generator **120**, an action manager **122**, one or more orders **124** and an order processing module **126**. Furthermore, in communication with the store server process **106** is a storage device such as a hard disk which contains HTML structures **128** which define the layout of different HTML pages. In addition, the store server process **106** is in communication with one or more databases **130** with a database module **132**.

Broadly speaking, the dynamic page generator **120** responds to consumer browser **110** requests for the HTML pages. The dynamic page generator **120** can generate customized HTML pages with the HTML structures **128**. For example, a merchant can define HTML pages which display the store's lobby, product images, product descriptions, and order forms.

The action manager **122**, on the other hand, executes various functions (also called actions) in response to consumer input. The orders **124** contain sales transaction information in a unique format, while the order processing module **126** processes the orders **124**.

When a consumer initiates a sales transaction, the consumer browser **110** sends sales transaction information to the store server process **106**. The store server process **106** represents the sales transaction information in each order **124** using key-value pairs. Preferably each order **124** stores the key-value pairs in one or more blackboards. The key-value pairs define different aspects of a sales transaction. For example, the key-value pairs define which items the consumer has selected, the number of desired items, where to ship the items, the identity of the consumer, the billing information, etc. As explained in more detail below, different components in the order processing module **126** process the key-value pairs to complete a sales transaction.

For example, assume that a merchant sells watches. When a consumer directs the consumer browser **110** on the consumer computer to access the merchant system **104**, the dynamic page generator **120** creates web pages with illustrate different watches offered for sale. As explained in more detail below, when the consumer selects a watch for purchase, the action manager **122** generates an order **124** and adds a number of key-value pairs to the order **124**. In this example, the key-value pairs contain sales transaction information associated with the selected watch.

For instance, the key-value pairs define information about the order **124** such as who the shopper is, where to send the watch, what is the means of payment, etc. In addition, the key-value pairs identify information about the watch such as the watch's identification code, the number of desired watches, the style and color or watch, etc. Because of the implementation of the unique order **124** in the present invention, different key value pairs can be easily added for different sales situations.

Unlike other order forms which rely on specific database structures or other structured formats, the order **124** of the present invention contains as many key-value pairs are necessary to define a sales transaction. This allows merchants to easily customize the electronic merchandising system **100** for diverse sales transactions. Advantageously, merchants can add new key-value pairs or delete existing key-value pairs to modify the electronic merchandizing system **100**.

III. Architecture Of The Preferred Electronic Merchandising System

As illustrated in FIG. 2, the preferred electronic merchandising system architecture contains one or more consumer clients 102, one or more merchant clients 200 and a merchant system 104. The merchant system 104 contains an Internet Information server 202, a router 204, a global configuration controller 206, one or more store server processes 106, multiple HTML structures 128 and one or more databases 130. In general, the consumer clients 102 and merchant clients 200 communicate with the merchant system 104 via the communications medium 108.

A. The Communications Medium

Focusing now on the communications medium 108 as shown in FIG. 2, the presently preferred communications medium 108 includes the Internet which is a global network of computers. The structure of the Internet, which is well known to those of ordinary skill in the art, includes a network backbone with networks branching from the backbone. These branches, in turn, have networks branching from them, and so on. For a more detailed description of the structure and operation of the Internet 33, please refer to *The Internet Complete Reference* by Harley Hahn and Rick Stout, published by McGraw-Hill, 1994.

One of ordinary skill in the art, however, will recognize that a wide range of computer networks can be employed in the present invention. For example, the communications medium 108 can include interactive television networks, telephone networks, wireless data transmission systems, two-way cable systems, customized computer networks, interactive kiosk networks, automatic teller machine networks, and the like.

In addition to the Internet, the communications medium 108 may also contain Internet providers. An Internet provider is a computer system which provides Internet access to the consumer computers. Examples of Internet providers include the Microsoft Network (MSN), America On-line, Prodigy, Compuserve, and Network Intensive, to name a few. Many users pay monthly access fees to the Internet providers because the Internet providers provide local telephone connections, a variety of help services and an organized format for accessing the Internet. The Internet providers are optional, and in some cases, the consumer clients 102, may have direct access to the Internet. For example, the consumer clients 102 may be connected to a local area network which in turn is directly connected to the Internet.

When a consumer desires to access information available on the Internet, the consumer initiates a connection from his or her consumer client 102. The consumer browser 110, in turn, establishes a communication link directly with the Internet or with the Internet provider via a communications link. Once connected to the Internet, the consumer can direct the consumer browser 110 to access the merchant system 104.

One popular part of the Internet is the World Wide Web. The World Wide Web contains different computers which store HTML documents capable of displaying graphical and textual information. The content providers which provide information on the World Wide Web are typically called "websites." A website is defined by an Internet address which has one or more HTML pages. Generally, an HTML page is an electronic document which organizes the presentation of text, graphical images, audio and video.

B. The Consumer Clients

Focusing now on the consumer clients 102 which include the consumer browsers 110, the consumer client 102 operates on a general purpose computer (hereinafter referred to as the consumer computer 102). In the preferred embodiment, the consumer computer 102 is a conventional personal computer which is equipped with a conventional modem. Preferably, the consumer computer 102 runs an appropriate operating system such as the Microsoft Windows® 3.1, Microsoft Windows® 95, Microsoft Windows® NT, the Apple® MacOS®, or IBM® OS/2® operating systems. As is conventional, the preferred operating system includes a TCP/IP stack which handles all incoming and outgoing message traffic passed over the communications medium 108.

In other embodiments, the consumer computer 102 could, for example, be a computer workstation, a local area network of computers, an interactive television, an interactive kiosk, a personal digital assistant, an interactive wireless communications device or the like which can interact with the communications medium 108. While in such systems the operating systems will differ, they will continue to provide the appropriate communications protocols needed to establish communication links with the communications medium 108.

In the preferred embodiment, the consumer browser 110 is a software program which allows a consumer to access the merchant system 104 over the communications medium 108. In the preferred embodiment, the consumer browser 110 is the Microsoft Internet Explorer version 3.0 developed by Microsoft Corporation. One of ordinary skill in the art, however, will recognize that numerous other types of access software could also be used to implement the present invention. These other types of access software could, for example, be other types of Internet browsers such as the Netscape Navigator developed by Netscape, Inc., or other types of client applications including custom network browsers, two-way communications software, cable modem software, point-to-point software and the like.

Associated with each consumer browser 110 is an optional cookie module (not shown) which stores a shopper identification code. The shopper identification code, as described in more detail below, uniquely identifies each consumer. A "cookie" is a file which stores information on the consumer computer for use by the merchant system 104. In the preferred embodiment, the consumer browser 110 stores the shopper identification code in the cookie so that the consumer does not need to retype the shopper identification code each time the consumer accesses the electronic merchandising system 100. As discussed in more detail below, the preferred embodiment includes the shopper identification code in the Uniform Resource Locator (URL). The specification for cookies is defined by Netscape Corporation and can be found at http://www.netscape.com/newsref/std/cookie_spec.html.

In an alternative embodiment, a cookie does not store the shopper identification code. Rather, as discussed in more detail below, the shopper identification code is obtained by the merchant system 104 from the database 130 when the consumer accesses the merchant system 104. In such embodiments, when the consumer accesses the merchant system 104, the merchant system 104 prompts the consumer to enter a password which the merchant system 104 then uses to obtain the shopper identification code from the database 130. The shopper identification code is then combined with the merchant system URLs as described below in more detail.

C. The Merchant Clients

Focusing now on the merchant client 200, the merchant client 200 allows a merchant to interact with the merchant system 104. In some embodiments, the merchant client 200 may execute on the same device as the merchant system 104. In other embodiments, the merchant client 200 executes on a separate computer which accesses the merchant system **104** via the communications medium **108**. In the preferred embodiment, a merchant uses the merchant client **200** to configure the merchant system **104**.

Preferably, the merchant client **200** executes on a conventional personal computer (the merchant computer **200**) which is equipped with a conventional modem. The merchant client **200** runs on an appropriate operating system such as the Microsoft Windows® 3.1, Microsoft Windows® 95, Microsoft Windows® NT, the Apple® MacOS®, or IBM® OS/2® operating systems. As is conventional, the preferred operating system includes a TCP/IP stack which handles all incoming and outgoing message traffic passed over the communications medium **108**.

In the preferred embodiment, the merchant client **200** includes a merchant browser **210**. The merchant browser **210** is a software program which allows a merchant to access the merchant system **104** over the communications medium **108**. In the preferred embodiment, the merchant browser **210** is the Microsoft Internet Explorer, Version 3.0 developed by Microsoft Corporation. One of ordinary skill in the art, however, will recognize that numerous other types of access software could also be used to implement the present invention. Such access software could, for example, be other types of Internet browsers such as the Netscape Navigator developed by Netscape, Inc., and other client software such as custom network browsers, two-way communications software, cable modem software, point-to-point software and the like.

D. The Merchant System

The computer facility associated with the merchant system **104** is called the merchant system computer 104. In the preferred embodiment, the merchant system **104** may exist on a single merchant system computer **104** or may be distributed across a cluster of merchant system computers **104**. The merchant system **104** includes an internet information server **202**, a router **204**, a global configuration controller **206**, at least one merchant store server **212**, a storage medium for HTML structures **128** and one or more databases **130**.

In the single computer configuration, the merchant system **104** includes a conventional computer which is equipped with a high speed communications link to the Internet. Preferably, the merchant system computer **104** is a general purpose Pentium class (or better) computer, which has at least 16 megabytes of random access memory and at least 45 megabytes of free hard disk space. The preferred operating system is Microsoft Windows® NT version 3.51 or later with a Windows® NT file system.

As discussed in more detail below, each computer in the merchant system **104** also has a registry which stores information about the merchant computer's hardware and software. Preferably, the registry is part of the computer operating system. In the preferred embodiment, the registry is the Windows® NT registry which is part of the Windows® NT operating system. The Windows® NT registry and the methods for modifying the Windows® NT registry are well known in the art, and are described in *The Win* 32 *Programmer's Reference Vol. II* Microsoft Press, 1993, pp. 203–239.

In addition, the computer which executes the merchant system **104** contains Service Pack 3 and the Microsoft Internet Information Server version 1.0 or later which are available from the Microsoft Corporation. The Microsoft Windows® NT operating system includes a TCP/IP stack which handles all incoming and outgoing message traffic passed over the communications medium **108**. The computers in the merchant system **104**, can, however, include a wide range of devices which provide information, graphics or text. These devices may contain specialized operating systems which communicate using their respective communications protocols.

In the distributed configuration, a cluster of computers execute the merchant system **104** and are interconnected via the variety of high speed local area networks (LAN) supported by the Windows® NT operating system. One of the merchant system computers **104** in the cluster of merchant system computers **104** is configured to store global configuration information and is called the global configuration controller **206**.

Preferably, the global configuration controller **206** contains global configuration information which defines the merchant system configuration. The global configuration information may, for example, include the names of all the electronic stores on the merchant store server **212**, the location of databases **130** within each store, the location of template files, system error messages, system constants such as date and currency formats, etc.

The Internet information server **202** is a World Wide Web server. The Internet information server **202** supports the use of virtual servers, allowing multiple web servers to run on a single computer. The Internet information server **202** also uses the HyperText Transmission Protocol (HTTP) to communicate with the consumer browsers **110** or the merchant browser **210**. The Internet information server **202** may be implemented using any one of a number of commercially available software packages, including the Internet Information Server (IIS) which is available from Microsoft Corporation.

Focusing now on the router **204**, the router **204** is an Internet Server Application Programming Interface (ISAPI) filter. ISAPI is a programming interface developed by, and available from, the Microsoft Corporation. The ISAPI filters use named pipes to connect the consumer browsers **110** with the store server processes **106**. Named pipes are interprocess communication methods which transfer data between separate processes and are well known to one skilled in the art. For more information on pipes, see Abraham Silberschatz and Peter B. Galvin, *Operating System Concepts* Addison-Wesley Publishing Company, 1994.

The router **204** examines the universal resource locator (URL) address specified in a consumer browser **110** or merchant browser request, and determines from the URL if the URL is a request which specifies one of the merchant store servers **212**. In the preferred embodiment, the router **204** and the merchant store server **212** utilize the global configuration information to interconnect the consumer browsers **110** with the store server processes **106**.

For example, when the router **204** receives a URL for a specific store, the router **204** uses the global configuration information stored in the global configuration controller **206** to locate the desired store server process **106**. The router **204** then routes requests from the consumer browser **110** to the desired store server process **106**. Once the store server process **106** executes the request, the store server process **106** sends the results back to the requesting consumer browser **110** via the router **204**.

One or more store server processes **106** exist in the merchant store server **212**. In the preferred embodiment, the components in the store server process **106** are written in the Python and C programming languages. The Python programming language is a portable, interpreted, object-oriented programming language developed at the Centrum voor Wiskunde en Informatica (CWI) and is available at http://www.python.org. The preferred C compiler is the Microsoft Visual C++ programming language which is available from Microsoft Corporation. One of ordinary skill in the art, however, will recognize that other software programming languages could be used to implement the present invention.

In the preferred embodiment, the store server processes 106 are also in communication with the storage device which stores the HTML structures 128. Furthermore, the store server processes 106 are in communication with one or more databases 130. Preferably, the databases 130 are Structured Query Language (SQL) databases 130. Each store server process 106 communicates with the databases 130 and with the database module 132 as illustrated in FIG. 3. Preferably, the database module 132 is a structured query server. The database module 132 creates the queries which access the databases 130. The architecture of the database module 132 is further described in a concurrently filed application having the title "Database Schema Independence", which is incorporated herein by reference.

The structured query language implemented in the database module 132 and the databases 130 is a standard defined by the International Standards Organization (ISO) for defining, updating and querying relational databases. For example, the databases 130 can be implemented with any number of commercial database programs including Microsoft Access, Oracle's relational database products and the like. The databases 130 may be either local to the store server processes 106, or may be accessible to the store server processes 106 over one or more conventional local area networks (LAN).

1. The Dynamic Page Generator

During a typical shopping session, the consumer browser 110 and the store server process 106 communicate with each other over the communications medium 108. Typically, the consumer browser 110 sends URL addresses to the store server process 106, and the store server process 106 responds with HTML documents. The HTML documents may contain registration information, product offerings, promotional advertisements, order forms, etc.

The dynamic page generator 120 generates the HTML documents sent to the consumer browser 110. The dynamic page generator 120 dynamically creates HTML documents in response to commands generated by the consumer browser 110. The commands generated by the consumer browser 110 utilize the standard GET/POST format of the HyperText Transport Protocol (HTTP). For example, as discussed in more detail below, the buttons or other content items in an HTML page contain a hyperlink to a URL. When the consumer selects the button within the consumer browser 110, the consumer browser 110 generates an HTTP GET message which includes the URL associated with the button. The HTTP GET message and the associated URL is then sent from the consumer Web browser to the dynamic page generator 120.

When the dynamic page generator 120 receives the HTTP GET message and the associated URL, the dynamic page generator 120 identifies an HTML structure 128 which is hereinafter referred to as the HTML templates 128. The dynamic page generator 120 then processes the HTML template 128 to generate the appropriate web page. The architecture of the dynamic page generator 120 and the HTML templates 128 are described in a concurrently filed application having the title "Electronic Shopping And Merchandising System" which is incorporated herein by reference. While the dynamic page generator 120 in the preferred embodiment dynamically generates the HTML pages based on commands which exist in the HTML template 128, in other embodiments, the dynamic page generator 120 may have a database of predefined HTML pages.

In the preferred embodiment, the consumer can navigate throughout the virtual store and select items for purchase by storing the selected items in a shopping cart. The merchant defines his virtual store with one or more HTML pages. While the preferred embodiment uses a number of HTML pages, the present invention is flexible and allows a merchant to configure his virtual store as desired. For example, in the preferred embodiment, the merchant defines the layout of the initial welcome page with a welcome.html template. The merchant defines the layout of the virtual store lobby with a main.html template. The merchant defines the layout of a sales department in the virtual store with the dept.html template. The merchant defines the layout of the product pages which display items for sale with the product.html template. The merchant can also define the layout of the shopping basket with the basket.html template, the layout of a displayed order form with an orderform.html template and the layout of the billing information with an accepthtml template.

For example, FIG. 4 illustrates the shopping cart HTML page 400. The shopping cart HTML page 400 includes the consumer browser menus 402, the URL address 404, and a number of option buttons. In this example, the option buttons include the geared up button 406, the base camp button 408, and the off the wall button 410 which allow a consumer to browse about the virtual store. In addition, the option buttons include a shopping cart button 412 which directs the electronic merchandising system 100 to display the shopping cart and a check out button 414 which directs the electronic merchandising system 100 to initiate a sales transaction. Displayed in the shopping cart is the list of items the consumer has selected.

In this example, the layout of the shopping cart HTML page 400 is defined with the basket.html template. As explained in more detail below, when the consumer selects the shopping cart button 412, the URL associated with the shopping cart button 412 directs the dynamic page generator 120 to access the basket.template. The dynamic page generator 120 then creates the shopping cart HTML page 400.

FIG. 5 illustrates the shipping and payment check out HTML page 500. A merchant defines the layout of the shipping and payment check out page 500 with the orderform.html template. In this example, the consumer selects the check out button 414 which directs the dynamic page generator 120 illustrated in FIG. 3, to display the check out HTML page 500. The check out HTML page 500 displays an order form with the list of selected items and a subtotal.

FIG. 6 illustrates the purchase summary check out HTML page 600 (hereinafter referred to as the acceptance HTML page) 600. A merchant defines the layout of the acceptance HTML page 600 with the accepthtml template. In this example, the acceptance HTML page 600 displays the subtotal, taxes, shipping costs and total amount of the sales transaction. Furthermore, the consumer can enter pertinent billing information such as the consumer's credit card information.

In the preferred embodiment, the dynamic page generator 120 and each HTML template is identified with a particular URL. The format of the preferred URL is "http://servername/environment.securitytype/componentname/

storename/shopper_ID/arguments parameters." For example, information entered by the consumer into the HTML page can be passed to the dynamic page generator with the arguments and parameters specified in the URL. In the preferred URL, the servername is the name of the WEB server host. The environment defines the store environment. Three types of store environments exist: 1) a production environment, 2) a development environment and 3) a post-development testing environment. A merchant uses the production environment for actual store operations. In the preferred embodiment, the production environment is identified with the "prd" acronym.

The security type identifies whether the store server process **106** is secure or insecure. A secure store server process **106** is identified with the "s" character and an insecure store server process **106** is identified with the "i" character. The componentname identifies the dynamic page generator **120**. The dynamic page generator **120** is identified with the "pgen" acronym. The storename identifies the name of a store. The shopper_ID is the shopper identification code which identifies each consumer. The arguments and parameters specify the values which are passed to the dynamic page generator **120**. For example, an argument can identify one of the HTML templates.

In this example, the URL for the detailed product HTML page identifies the servername, uses the prd acronym to identify that the production environment is for an actual store, identifies that the store server process **106** is insecure with the "i" character, identifies the dynamic page generator **120** with the "pgen" acronym, designates the storename, designates the shopper identification code, the product.html template, and the stock keeping unit (sku) of the desired product as follows: http://servername/prd.i/pgen/storename/shopper$_{13}$ id/product.html sku=7777.

When a consumer selects one of the products, the consumer browser **110** sends the product's URL with the products sku to the merchant system **104**. The merchant system **104** then forwards the URL to the dynamic page generator **120**. In this example, the dynamic page generator **120** receives the URL and creates the product HTML page identified by the product.html template and the product store keeping unit. As explained in more detail below, the dynamic page generator **120** also obtains data about the product from the order processing module **126**. The dynamic page generator **120** then combines the product data with the product.html template to create the product HTML page.

In the preferred embodiment, the product data and other merchandising data is stored in a number of tables located in the database **130**. The tables necessary to enable the preferred embodiment include, but are not limited to, a shopper table **300**, a product table **302**, an order table **304**, a department table **306**, a receipt table **308** and a promotion table **310**.

FIG. 7 illustrates the format of the preferred shopper table **300**. Each row in the shopper table **300** corresponds to a particular consumer (shopper) while each column contains information related to the shopper identification code. For example, the preferred shopper table **300** contains columns with the shopper identification code, date the shopper identification code was created, member data, the consumer's name, address, city, state, zip code, country, etc. The shopper table **300** is merchant defined and can include a wide variety of consumer information such as columns for the consumer's password, the customer's size and the customer's preferences. The merchant specifies the location of the shopper table **300** in the registry.

Associated with the shopper table **300** is the shopper manager **320**. The shopper manager **320** adds, modifies and deletes the entries existing in the shopper table **300**. In the preferred embodiment, the shopper manager **320** is an object which uses well-known database techniques for adding, modifying and deleting the entries in the shopper table **300**. Other components in the electronic merchandising system **100** rely on the shopper manager **320** to access the rows in the shopper table **300** and retrieve, store or modify the consumer information stored in the shopper table **300**. In addition, the shopper manager **320** generates and assigns the shopper identification codes which are stored in the shopper table **300** and which uniquely identify each consumer. In the preferred embodiment, the shopper identification code is a 128-bit globally unique identifier (GUID).

The product tables **302** associated with the preferred embodiment are illustrated in FIGS. **8**A and **8**B. In the preferred embodiment, two product tables **302** exist—a product family table **800** and a product variant table **802**. These product tables **800** and **802** are merchant defined and can include a wide variety of information. The preferred product family **800** table is merchant defined and stores information about a product family. Each row in the product family table **800** is a record corresponding to a particular product family while each product family table column contains information related to the product families. The columns contain product information such as the product family identifier, the product family name, the product family description, the department identifier, the size type, the data introduced, the list price, the sale price, the sale start, the sale end, the image file name, etc. The merchant specifies the location of a query which queries the product variant table **802** in the registry.

The product variant table **802** is also merchant defined and stores information for a specific product within the product family. Each row in the product variant table **802** is a record corresponding to a particular product while each column contains information related to the products. For example, the product variant columns may contain a product's family identifier, stock keeping unit (sku), a color value, a size value, etc. The format of the product family table **800** is merchant defined and can contain wide variety of product characteristics. The merchant specifies the location of a query which queries the product variant table **802** in the registry.

The preferred order table **304** is illustrated in FIG. **9**. The preferred order table **304** stores items which the consumer has selected for purchase. Each row corresponds to an order while each column contains information related to items in the order. The order table **304** is merchant defined and can include a wide variety of order information. For example, the order table **304** contains the order identification code, the shopper identification code, the date the consumer last modified the order table **304**, the key-value pairs associated with the order **124**, etc. The location of the order table **304** is specified in the registry.

Associated with the order table **304** is an order manager **322**. The preferred order manager **322** adds, modifies and deletes the entries existing in the order which are also stored in the order table **304**. In the preferred embodiment, the order manager **322** is an object which uses well-known database techniques for adding, modifying and deleting the entries in the order table **304**. Other components in the electronic merchandising system **100** rely on the order manager **322** to access the rows in the order table **304** and retrieve, store or modify the order information stored in the order table **304**. In addition, the order manager **322** generates and assigns the order identification code which uniquely identifies each shopper. In the preferred embodiment, the order identification code is a 128-bit globally unique identifier (GUID).

The preferred department table **306** stores information about individual sales departments in the virtual store. Each row corresponds to a particular department while each column contains information related to the department. For example, the columns in the department table **306** contain the department identifier which identifies a department, the department name and the department description. The department table **306** is merchant defined and can include a wide variety of department information. The location of the department table **306** is specified in the registry.

The receipt table **308** is only required if receipts are provided in the virtual store. Each row in the receipt table **308** corresponds to a particular receipt, while each column contains information related to the receipt. The format of the receipt table **308** is merchant defined and can include a wide variety of receipt information. The location of the receipt table **308** is specified in the registry.

The preferred promotion table **310** defines different promotions offered in a virtual store. Each row corresponds to a particular promotion, while each column contains information related to the promotion. The format of the promotion table **310** is merchant defined and can include a wide variety of promotion information. For example, the preferred promotion table **310** contains columns with the promotion identifier which uniquely identifies a promotion, a promotion name, a promotion description, a promotion rank, an active code which indicates whether the promotion is active, the date when a promotion starts, the date when a promotion ends, whether the promotion is available to all consumers, etc. The location of the promotion table **310** is specified in the registry.

2. The Action Manager

In the preferred embodiment, the action manager **122** is an object which contains a number of different actions. An action is a routine which performs specific functions. The pertinent actions needed to process an order **124** in the preferred embodiment are illustrated in FIG. 3. The actions include: an order.additem action **330**, an order.clearform action **332**, an order.clearitems action **334**, an order.deleteitem action **336**, an order.deleteorder action **338**, an order.edititem action **340**, an order.editorder action **342**, an order.editquantities action **344**, an order.plan action **346**, and an order.purchase action **348**.

Each of the actions in the action manager **122** is identified by a URL. The arguments which are passed to the action manager are identified in the URL. In the preferred embodiment, the format of the URLs which designate a particular action are similar to the URLs which designate the dynamic page generator **120** and have the following format: "http://servername/environment.securitytype/componentname/storename/shopper_ID/module.action;arguments."

In the preferred URL, the servername is the name of the WWW server host. The environment defines the store environment The security type identifies whether the store server process **106** is secure or insecure. The componentname identifies the action manager **122** with the "xt" acronym. The storename identifies the name of a store. The shopper_ID is the shopper identification code which identifies each consumer. The module.action identifies the desired module and action. The arguments are values which are passed to the desired module and action. For example, an argument might specify a product's sku, the consumer's shipping address or the consumer's billing information, etc.

For example, if a consumer views a product in a product HTML document, the consumer can add the product to his shopping cart by selecting a button. The button in the HTML document specifies a URL which invokes the order.additem action **330** in the action manager **122**. The format of the URL is "http://servername/prd.i/xt/storename/shopper _ID/order.additem;sku=1234." In this example, the URL specifies the action manager **122** with the "xt" acronym, the order.additem action **330** and the product's store keeping unit (sku). As discussed in more detail below, when the action manager **122** receives the URL, the action manager **122** executes the order.additem action **330**.

Focusing now on the preferred format and architecture of the order actions, the order.additem action **330** adds an item to an order **124**. The URL which invokes the order.additem action **330** includes an argument which specifies the sku of a particular item to add to the order **124**. In the preferred embodiment, the order.additem action **330** creates an order **124** with key-value pairs about the added item and passes the order **124** to the components in the order processing module **126**. The URL which invokes the order.additem action **330**, contains the product sku or the location where the order.additem action **330** can obtain the product sku. In addition, the URL can contain the number of items (quantity) and the price of the items.

As described in more detail below, the preferred order **124** is an object which stores the key-value pairs. The order.additem action **330** instantiates the order **124** with well-known object-oriented techniques which allocate enough memory to hold the order **124**. After instantiating the order **124**, the order.additem action **330** adds the sku key-value pair and other initial key-value pairs as discussed in more detail below to the order **124**. The order.additem action then passes the order **124** to the order processing module **126**.

As explained in more detail below, the order processing module **126** processes the order **124** and adds key-value pairs containing product information about the selected item to an item blackboard in the order **124**. Furthermore, the URL which invokes the order.additem action **330** can also identify which HTML page to display after completion of the order.additem action **330**. In the preferred embodiment, the default configuration of the order.additem action **330** directs the dynamic page generator **120** to display the shopping cart HTML page **400** after obtaining the product information from the order **124**. A merchant can, however, direct the order.additem action to display other HTML pages. When generating the shopping cart HTML page **400**, the dynamic page generator **120** combines the order information in the order table **304** with the shopping cart HTML page **400**.

The order.clearform action **332** deletes all the items and all the properties associated with an order **124** in the order table **304**. Typically, the URL which specifies the order.clearform action **332** is associated with a clear order button or menu option in the HTML pages. When invoked, the order.clearform action **332** directs the order manager **322** to delete the order **124** from the order table **304**. In the preferred embodiment, the default configuration of the order.clearform action **332** directs the dynamic page generator **120** to display the shopping cart HTML page **400** after deleting the order **124** from the order table **304**.

The order.clearitems action **334** deletes all the items in the order table **304** which are associated with a particular order **124**. However, the order properties (information which relates to the entire sales transaction such as the shipping information, billing information, the shopper identification code, etc.) remains in the database **130**. Typically, the URL which specifies the order.clearitems action **334** is associated with a clear items button or menu option in the HTML pages. In the preferred embodiment, the default configuration of the order.clearitems action **334** also directs the dynamic page generator **120** to display the shopping cart HIML page **400** after clearing the item information in the order table **304**.

The order.deleteitem action **336** deletes an item in the order table **304**. The URL which invokes the order.deleteitem action **336** specifies an index which identifies a particular item to delete. Typically, the URL which specifies the order.deleteitem action **336** is associated with a delete item button or menu option in the HTML pages. In the preferred embodiment, the order.deleteitem action **336** default configuration also directs the dynamic page generator **120** to display the shopping cart HTML page **400** after deleting an item from the order table **304**.

The order.deleteorder action **338** deletes an entire order **124** from the order table **304** by directing the order manager **322** to delete the entries associated with an order **124** in the order table **304**. Typically, the URL which specifies the order.deleteorder action **338** is associated with a delete order button or menu option in the HTML pages. In the preferred embodiment, the order.deleteorder action **338** default configuration also directs the dynamic page generator **120** to display the shopping cart HTML page **400** after deleting the order **124**.

The order.edititem action **340** is similar to the order.additem action **330**, but modifies an existing item in the order **124** rather than adding a new item. The URL which invokes the order.edititem action **340** contains in the URL arguments 1) an index which identifies the item to edit and 2) arguments which specify how to modify the item. For example, the URL may contain an item index and a quantity value which specifies the new quantity of desired items. In the preferred embodiment, the order.edititem action **340** default configuration also directs the dynamic page generator **120** to display the shopping cart HTML page **400** after modifying an item in the order table **304**.

The order.editorder action **342** adds additional order properties to the order **124**. Typically, the order.editorder action **342** adds shipping and billing information to the order **124**. The order.editorder action **342** can also be modified to add custom key-value pairs. The URL which invokes the order.editorder action **342** can contain a wide variety of properties which the order.editorder action **342** adds to the order table **304**. In the preferred embodiment, the order.editorder action **342** default configuration directs the dynamic page generator **120** to display the shipping and billing check out HTML, page **500**.

The order.editquantities action **344** modifies the quantity of items in the order **124**. The URL which invokes the order.editquantities action **344** specifies the items to be modified. Typically, the order.editquantities action **344** then directs the order manager **322** to modify the quantity column in the order table **304** for a specified items. In the preferred embodiment, the order.edititem action **340** default configuration also directs the dynamic page generator **120** to display the shopping cart HTML page **400** after modifying the item quantity.

The preferred order.plan action **346** processes the order **124** to compute the total cost of the items in the order **124**. As explained above, the URL which invokes the order.plan action **346** includes arguments relating to the order **124**. The order.plan action **346** creates an order **124** and adds the arguments passed to the order.plan action in the URL are added as key-value pairs to the order **124**. The order.plan action **346** then passes the order **124** to the order engine in the order processing module **126**. The order engine then invokes the components in the order pipeline which process the order **124** and determines the total cost of the selected items. Preferably, the order.plan action **346** default configuration then directs the dynamic page generator **120** to display the total cost of the selected items in the acceptance HTML page **600**. The consumer can then confirm or abort the sales transaction.

The order.purchase action **348** completes the purchase process. The URL which invokes the order.plan action **346** includes arguments relating to the order **124**. The order.purchase action **348** is similar to the order.plan action **346** in that it creates an order **124** and adds the arguments in the URL as key-value pairs to the order **124**. The order.purchase action **348** then passes the order **124** to the order engine in the order processing module **126**. The order engine then executes each component in the order pipeline to complete the sales transaction.

3. The Order

A conventional order form **1000** is illustrated in FIG. **10**. Such an order form typically contains a list of purchased items **1002** and other order information. The order information, also referred to as the order properties, contain a date **1004**, the name of the shopper **1006**, the shopper's address **1008**, the shipping address **1010**, the billing information **1012**, the order subtotal **1014**, the taxes **1016** and the order total **1018**. The list of items **1002** typically contains an entry for each item and item information such as the quantity of items, the color, size and model of items, the item discount, item price, etc. The format of such order forms **1000** will vary from merchant to merchant. For example, an international merchant may require complex tax information, a merchant which provides unique billing options will need different billing information, a merchant which provides gift wrapping will need gift wrapping information, etc.

The electronic merchandising system **100** of the preferred embodiment, however, can be flexibly modified for any type of sales transaction. Rather than utilizing a predefined structure of data elements, the present invention utilizes an unorganized format of key-value pairs. The key-value pairs can be easily added to or deleted from an associative array called a blackboard. As shown in FIG. **3**, in the preferred embodiment, the order contains an order blackboard **350** and one or more item blackboards **352**.

A detailed diagram of the order **124** is illustrated in FIG. **11**. Unlike organized database entries, the key-value pairs **1100** in the blackboards **350** and **352** are indexed by keys. Each key is a string which uniquely identifies its associated value. To locate a particular value, the present invention searches one of the blackboards **350** or **352** for the proper key and then accesses the value associated with the key.

In the preferred embodiment, the order **124** is an object which comprises at least one order blackboard **350** and zero or more item blackboards **352**. Preferably, each blackboard contains a key and a value for each key-value pair **1100**. The key-value pairs **1100** in the order blackboard **350** contain order properties such as the order date **1004**, the consumer's name **1006**, the consumer's address **1008**, the desired shipping address **1010**, the billing information **1012**, the order subtotal **1014**, the taxes **1016**, the order total **1018**, etc.

The preferred format of the order key-value pairs **1100** is "order.key." For example, the key for the order identification code in the order blackboard **350** can be represented as "order.order_id." The "order" identifies the order blackboard **350** while the "order_id" designation identifies the order_id key-value pair.

The key-value pairs **1100** in the item blackboards **352** contain item information. Preferably, an item blackboard **352** exists for each item. Furthermore, the key-value pairs **1100** in one item blackboard **352** can differ from the key-value pairs **1100** in another item blackboard **352**. The preferred format of the item key-value pairs **1100** is "item.key." For example, the key for an item's stock keeping unit (sku) can be represented as "item.sku" where "item" identifies the item blackboard **352** and the "sku" designation identifies the sku key-value pair.

When an order **124** is instantiated, a number of initial key-value pairs **1100** are added to the order blackboard **350** and one or more item blackboards **352**. The initial key-value pairs **1100** include, but are not limited to: an order.order_id key-value pair, an order.shopper_id key-value pair, an order.messages key-value pair, an item.sku key-value pair, an item.quantity key-value pair, an item.placed_price key-value pair and in addition, other key-value pairs used by the components in the order pipeline for computational purposes.

In the order blackboard **350**, the order_id value in the order_id key-value pair contains the order identification code which uniquely identifies each order **124**. The shopper_id value in the shopper_id key-value pair contains the shopper identification code which uniquely identifies each shopper. The messages value in the order.messages key-value pair identifies the language used for error messages. The messages value is initially set by a value from the registry which is typically "USA."

In each item blackboard **352**, the sku value in the sku key-value pair identifies the sku which uniquely identifies a particular item. The quantity value in the quantity key-value pair identifies the number of ordered items. The placed_price value in the placed_price key-value pair identifies the price of the ordered item. The oadjust_adjustedprice value in the oadjust_adjustedprice key-value pair identifies the amount of the adjusted order price. The n_unadjusted value in the n_unadjusted key-value pair is the number of items which have not been adjusted.

As discussed in more detail below, the components in the order processing module **126** modify some of the initial key-value pairs **1100** and add new key-value pairs **1100** to the order blackboard **350** and item blackboards **352** when processing the order **124**.

4. The Order Processing Module

As illustrated in FIG. **3**, the order processing module **126** contains an order engine **360** and an order pipeline **362**. The order pipeline **362** contains multiple stages which process the order **124**. The order engine **360**, on the other hand, is an object which invokes one or more of the stages in the order pipeline **362**. The stages in the preferred order pipeline **362** include the product information stage **364**, the merchant information stage **366**, the shopper information stage **368**, the order initialization stage **370**, the order check stage **372**, the item price adjust stage **374**, the order price adjust stage **376**, the shipping stage **378**, the handling stage **380**, the tax stage **382**, the order total stage **384**, the inventory stage **386**, the payment stage **388** and the accept stage **390**.

Each stage has one or more components which process the key-value pairs in the order **124**. As illustrated in FIGS. 12A–12C, each stage preferably contains three component categories: 1) a default component, 2) an optional component and 3) a required component. The default component associated with a particular stage executes when the merchant has not specified any optional components for the particular stage. The optional components are components which exist in the preferred embodiment, components which a merchant creates, or components which a merchant purchases from a third party. The optional components are typically customized for different types of sales transactions and replace the default components when installed. The required component for each stage is executed to ensure system consistency.

a) The Product Information Stage

The first stage in the order pipeline **362** is the product information stage **364**. The components in the product information stage **364** retrieve product data from a database and write the product data to the item blackboards **352** in the order **124**. The preferred product information default component **1200** accesses the sku key-value pair in the item blackboard **352** and performs a database query that retrieves product data corresponding to the sku. The product information default component **1200** then stores the product data into the product key-value pairs existing in the item blackboard **352**.

After executing the database query, the product information default component **1200** stores the returned data in product key-value pairs existing in the item blackboard **352**. The preferred embodiment creates the keys for the item blackboard by obtaining the name of each column in the product tables and adding a "product_" prefix to the column name. For example, the "list_price" column becomes the "product_list_price" key. The actual list price amount is then stored in the value identified by the product_list_price key. The product key-value pairs in the item blackboard **352** can include, but are not limited to, a product_retail_price key-value pair, a product_list_price key-value pair, a product_name key-value pair, a product_in_stock key-value pair, a product_delete_field key-value pair, a product_currentprice key-value pair, a product_price key-value pair, a product_sale_start key-value pair, a product_sale_end key-value pair, a product_sale_price key-value pair, a product_local_inventory key-value pair, a product_weight key-value pair, etc. However, if the product query does not return information about a product, such as when a product has been deleted from the product tables, the product information default component **1200** sets the product_delete_field key-value pair associated with the deleted product to 1.

The configuration information for the product information default component **1200** is stored in the registry value product_query_name. The product_query_name defines a database query which obtains product information from the product tables **302**. The database query is a standard structured query language query which accesses the product tables **302** and obtains product information about one or more items.

In the preferred embodiment, a product information optional component **1202** does not exist. However, in other embodiments a merchant could write a product information optional component, for example, to retrieve product information from a legacy database. Thus, the product information default component **1200** passes the order **124** to a product information required component **1204**.

The preferred product information required component **1204**, obtains the order **124** and deletes any items with a product_delete_field key-value pair which is set to 1. This ensures that subsequent components will not process any deleted items, such as when an item is no longer offered by a particular store.

b) The Merchant Information Stage

In the preferred embodiment, the merchant information stage **366** is not implemented and thus does not contain a merchant information default component **1206**, a merchant information optional component **1208** or a merchant information required component **1210**. In other embodiments, however, the components **1206**, **1208** and **1210** in the merchant information stage **366** may be configured to retrieve merchant information from a merchant database and store the merchant data as key-value pairs in the order blackboard **350**.

c) The Shopper Information Stage

The components in the shopper information stage **368** sets the shopper key-value pairs in the order blackboard **350** by retrieving shopper information from the shopper table **300**. A preferred shopper information default component **1212** accesses the shopper_id key-value pair in the order blackboard **350** and performs a database query which accesses the identified shopper in the shopper table **300**. Each column in the shopper table **300** is then added to a shopper key-value pair in the order blackboard **350**.

For example, the shopper key-value pairs added to the order blackboard **350** can include the shopper_id key-value pair, a shopper_address key-value pair, a shopper_size key-value pair, a shopper_preferences key-value pair, etc. The configuration information for the shopper information default component **1212** is stored in the registry value shopper_query_name. The shopper_query_name defines a database query which obtains shopper information based on the shopper identification code from the shopper table **300**.

In the preferred embodiment, neither a shopper information optional component **1214** nor a shopper information required component **1216** exists. However, in other embodiments a merchant could write a shopper information optional component **1214**, for example, to retrieve shopper information from a legacy database.

d) The Order Initialization Stage

The components in the order initialization stage **370** obtains order information existing the order table **304** so that the order information is available to the components in the order pipeline **362**. The preferred embodiment does not have an order initialization default component **1218** or an order initialization required component **1222**. However, two order initialization optional components **1220***a* and **1220***b* do exist. The order initialization optional components **1220** include a CopyData component **1220***a* and a SetData component **1220***b*.

The CopyData component **1220***a* copies data to the key-value pairs. For example, the CopyData component **1220***a* can be used to initialize the shipping address from the shopper information. The configuration information for the CopyData component **1220***a* is stored in the registry as "WLStdOrder.CopyData when from_prefix to_fprefix, field1, field2 . . . fieldn" where the "when" argument specifies that data will only be copied when the target key-value pairs are blank, the "from_prefix" argument identifies the source key-value prefixes, the "to_prefix" argument identifies the target key-value prefixes, and the "field1, field2 . . . fieldn" arguments identify the key-value suffixes. For example, to copy the shipping address to the billing address, the command "WLStdOrder.CopyData copycheckbox ship_to_bill_to_name" directs the CopyData component **1220***a* to copy a ship_to_name key-value pair, to a bill_to_name key-value pair when a checkbox has been checked by the consumer.

The SetData component **1220***b* sets the value of any key-value pair. The configuration information for the SetData component **1220***b* is stored in the registry as "WLStdOrder.SetData name.value string" where the "name.value" argument specifies the key and the "string" argument identifies value to be placed in the key-value pair. For example, to set the consumer identification code in the order blackboard, a merchant enters the command "WLStdOrder.SetData order.shopper_id 123456L" which directs the SetData component **1220***b* to set the value identified by the shopper_id key to "123456L."

e) The Order Check Stage

The components in the order check stage **372** verify that the required information is present and may be used to modify the data before any further processing occurs. The preferred embodiment does not contain an order check default component **1224**. The preferred embodiment, however, does contain two order check optional components **1226** which are called an OrderValidate component **1226***a* and an OrderItemValidate **1226***b* component.

The OrderValidate component **1226***a* is configured to check the order for required data, and verify that the required key-value pairs exist. The configuration information for the OrderValidate component **1226***a* is stored in the registry as "WLStdOrder.OrderValidate validation string" where the "validation string" argument specifies a rule about the order property formats such as, for example, that a particular key must exist, the value associated with a key must have a number, the value associated with a key is a date, string, a minimum value or maximum value, etc.

The OrderItemValidate component **1226***b* is configured to check the order **124** for required items, and verify that the required items exist. The configuration information for the OrderItemValidate component **1226***b* is stored in the registry as "WLStdOrder.OrderItemValidate validation string" where the "validation string" argument specifies a rule about the properties associated with each item, such as, for example, that a particular key must exist, the value associated with a key must have a number, date, string, a minimum value or maximum value, etc.

A preferred order check required component **1230** checks to assure that there is at least one item in the order **124**.

f) The Item Price Adjust Stage

The item price adjust stage **374** calculates the regular and current prices of an item. The preferred item price adjust stage **374** does not contain an item price adjust default component **1232**.

However, the item price adjust stage **374** does contain two item price adjust optional components **1234** which are called a SaleAdjust component **1234***a* and an ItemPromo component **1234***b*. The SaleAdjust component **1234***a* accesses the product_sale_start, the product_sale_end and the product_sale_price key-value pairs and adjusts the regular and current prices of an item by setting the iadjust_regularprice and the iadjust_currentprice key-value pairs on the item blackboard **352**.

For each item, the SaleAdjust component **1234***a* sets the iadjust_currentprice key-value pair to the product_sale_price key-value pair if the current date is between the product_sale_start key-value pair and the product_sale_end key-value pair. The configuration information for the

OrderValidate component **1226***a* is stored in the registry as "WLStdOrder.SaleAdjust."

The ItemPromo component **1234***b* is optional and applies a promotional price adjustment to an item based on the product information. The preferred ItemPromo component **1234***b* is further described in a concurrently filed application having the title "Electronic Promotion System For An Electronic Merchant System" which is incorporated herein by reference. The ItemPromo component **1234***b* defines the type of item which receives a discount, the amount of the discount, the date the promotion begins and the date the promotion ends. If a promotion applies, the ItemPromo component **1234***b* stores the sale price in the product_current_price key-value pair.

The preferred item price adjust required component **1236** sets the value in the iadjust_regularprice key-value pair to the value stored in the product_list_price key-value pair if the iadjust_regularprice key-value pair is not already set. In addition, the preferred item price adjust required component **1236** sets the value in the iadjust_currentprice key-value pair to the value stored in the product_list_price key-value pair if the iadjust_currentprice key-value pair is not already set.

g) The Order Price Adjust Stage

The components in the order price adjust stage **376** set the adjusted price of each item in the order **124**. The preferred embodiment does not contain an order price adjust default component **1238**. The preferred embodiment, however, does contain an order price adjust optional component **1240** which is also called a DbOrderPromo component **1240**. The DbOrderPromo component **1240** is further described in a concurrently filed application having the title "Electronic Promotion System For An Electronic Merchant System" which is incorporated herein by reference. The DbOrderPromo component **1240** discounts the order amount based on the shopper, the items purchased, the discount award given, etc. The DbOrderPromo component **1240** stores the discounted amount in the oadjust_adjustedprice key-value pair for each item in the order **124**.

The order price adjust required component **1242** completes the order price adjustment by setting the oadjust_adjustedprice key-value pair to the current price if not already set. In addition, the order price adjust required component **1242** also sets the oadjust_subtotal key-value pair on the order blackboard **350** equal to the sum of all oadjust_adjustedprice key-value pairs associated with each item.

h) The Shipping Stage

The components in the shipping stage **378** calculate the total shipping charge. The shipping default component **1244** sets the shipping charges to zero by setting the value in the shipping_total key-value pair existing in the order blackboard **350** to zero. Three shipping optional components **1246** exist which a merchant may add to perform shipping calculations: 1) the FixedShipping component **1246***a,* 2) the LinearShipping component **1246***b* and 3) the TableShipping component **1246***c*.

The FixedShipping component **1246***a,* evaluates the shipping method in the shipping_method key-value pair and charges a fixed shipping amount by setting the shipping_total key-value pair to a fixed amount. The merchant adds the FixedShipping component **1246***a* to the registry with the command "WLStdOrder.FixedShipping method price" where the "method" argument specifies a shipping method and the "price" argument specifies a shipping charge. For example, to charge a fixed fee of $10.00 dollars for shipping an item via Federal Express, the merchant enters "WLStdOrder.FixedShipping FedEx 1000.

The FixedShipping component **1246***a* then evaluates the shipping_method key-value pair to determine whether Federal Express delivery has been selected. If so, the FixedShipping component **1246***a* adds a $10.00 fee to the shipping_total key-value pair.

The LinearShipping component **1246***b* relies on a rate computed with a basis value that the merchant specifies in the registry. The basis is multiplied by the rate to determine the shipping cost. The basis value is typically some attribute of the item, such as quantity or weight. The format of the registry command is "WLStdOrder.LinearShipping method basis rate" where the "method" argument identifies a particular shipping method, the "basis" argument is the name of the key-value pair to use as the basis (such as the quantity, currentprice, adjustedprice, or weight), and the "rate" argument is the number to multiply by the basis to obtain the price.

For example, assume the merchant desires to charge 20 cents for shipping each item via the United State Postal Service. In this example, the merchant enters into the registry "WLStdOrder.LinearShipping USMail quantity 20." Furthermore, assume that a consumer buys 2 units of one item and 4 units of another and specifies delivery via the United States Postal Service. In this example, the LinearShipping component **1246***b* will evaluate the quantity key-value pairs for each item and determine that the consumer has purchased six items. The LinearShipping component **1246***b* then multiplies the total number of items (six) by 0.20 cents to calculate $1.20 shipping amount. The LinearShipping component **1246***b* then stores the shipping amount in the shipping_total key-value pair in the order blackboard **350**.

The TableShipping component **1246***c* uses a lookup table to determine what the shipping cost should be. The merchant specifies a basis (the unit of measurement for the product), a rate per basis unit, a database query name, and optionally a key-value pair used to calculate the shipping cost. The specified database query searches the database for the proper value. The format of the TableShipping command in the registry is "WLStdOrder.TableShipping method basis queryname location" where the "method" argument identifies a shipping method, the "basis" argument identifies the key-value pair used to compute the shipping cost, the "queryname" argument identifies a database query, the "location" argument identifies the key-value pair used for the shipping calculation. If the location key-value pair is not specified, it defaults to the ship_to_zip key-value pair.

The method name, the basis, and the value of the location field are used to create the database query. The database query then uses well-known database techniques to obtain the corresponding shipping cost. The shipping cost is then stored in the shipping_total key-value pair.

The shipping required component **1248** verifies whether the shipping_total is set. If not, the shipping required component **1248** generates an error message. The error message is a string which is stored in the order **124**.

i) The Handling Stage

The components in the handling stage **380** calculate the total handling charge for the order **124**. The handling default component **1250** sets the handling charges to zero by setting the value in the handling_total key-value pair in the order blackboard **350** to zero. Three handling optional components **1252** exist which a merchant may add to perform handling calculations: 1) the FixedHandling component

1252*a*, 2) the LinearHandling component 1252*b*, and 3) the TableHandling component 1252*c*.

The FixedHandling component 1252*a* charges a fixed handling fee. In particular, the FixedHandling component 1252*a* evaluates the handling method in the handling_method key-value pair and charges a fixed handling amount by setting the handling_total key-value pair to the fixed amount. The merchant adds the FixedHandling component 1252*a* to the registry with the command "WLStdOrder.FixedHandling method price" where the "method" argument defines the type of handling and the "price" argument defines the handling cost. For example, to charge a fixed handling fee of $10.00 dollars the merchant enters into the registry "WLStdOrder.FixedHandling handling 1000."

The FixedHandling component 1252*a* then evaluates the handling_method key-value pair to determine whether the specified handling service has been selected. If so, the FixedHandling component 1252*a* adds a $10.00 fee to the handling_total key-value pair.

The LinearHandling component 1252*b* relies on a rate computed with a basis value that the merchant specifies in the registry. The basis is multiplied by the rate to determine the handling cost. The basis value is typically some attribute of the item, such as quantity or weight. The format of the registry command is "WLStdOrder.LinearHandling method basis rate" where the "method" argument identifies a particular handling method, and the "basis" argument is the name of the key-value pair to use as the basis (such as the quantity, product_currentprice, product_weight), and the "rate" argument is the number to multiply by the basis to obtain the price.

For example, assume the merchant desires to charge 50 cents for handling each item. In this example, the merchant enters the following command into registry: "WLStdOrder.LinearHandling handling quantity 50." Furthermore, assume that a consumer buys 2 units of one item and 4 units of another. In this example, the LinearHandling component 1252*b* will evaluate the quantity key-value pairs to determine that six items have been purchased. The LinearHandling component 1252*b* will then multiply the total number of items (six) by 0.50 cents to calculate a $3.00 handling amount. The LinearHandling component 1252*b* then stores the handling amount in the handling_total key-value pair.

The TableHandling component 1252*c* uses a lookup table to determine what the handling cost should be. The merchant specifies a basis (the unit of measurement for the product), a rate per basis unit, a database query name, and optionally a key-value pair used to calculate the handling cost. The specified database query searches the database for the proper value. The format of the TableHandling command in the registry is "WLStdOrder.TableHandling method basis queryname location" where the "method" argument identifies a shipping method, the "basis" argument identifies the key-value pair used to compute the shipping cost, the "queryname" argument identifies a database query, and the "location" argument identifies the key-value pair use for the handling calculation.

If the handling method is specified in the order 124, the TableHandling component 1252*c* uses the method, the basis, and the value of the location field to generate a database query. The database query uses well-known database techniques to determine the corresponding handling cost. The TableHandling component 1252*c* then stores the handling cost in the handling_total key-value pair.

The handling required component 1254 verifies whether the handling_total key-value pair contains a value. If not, the handling required component 1254 generates an error message.

j) The Tax Stage

The component in the tax stage 382 compute the total tax for a given order 124. The default tax component 1256 sets the tax_total key-value pair in the item blackboard 352 and the tax_total key-value pair in the order blackboard 350 to zero.

The tax optional components 1258 includes tax components for USA, Japan, Canada and Europe. The optional tax components 1258 are called the SimpleUSTax component 1258*a*, the SimpleJapanTax component 1258*b*, the SimpleCanadaTax component 1258*c* and the SimpleVATTax component 1258*d*. The SimpleUSTax component 1258*a* applies a state tax rate. The format of the SimpleUSTax command in the registry is "Simpletax.SimpleUSTax state:rate state:rate . . ." where the "state" argument is the name of the state where the purchased items are to be shipped and the "rate" argument is the state tax rate. For example, if the state of California has a tax rate of 8.5% and the state of Nevada has a tax rate of 4.0%, the SimpleTax component 1258*a* command in the registry is "Simpletax.SimpleUSTax CA:8.5 NV:4.0."

The SimpleUSTax 1258*a* component then evaluates the ship_to_country key-value pair to determine whether the purchased items are destined for the United States of America If so, the SimpleUSTax 1258*a* component evaluates the name of the state in the ship_to_state key-value pair and applies the specified tax rate to the order subtotal stored in the order_subtotal key-value pair. The SimpleUSTax 1258*a* component then stores the amount of additional taxes for each item in a tax_total key-value pair in each item blackboard 352. In addition, the SimpleUSTax 1258*a* component sums the additional taxes for each item and stores the total amount of additional taxes in an order tax_total key-value pair in the order blackboard 350.

The SimpleJapanTax component 1258*b* calculates a tax rate for Japan. The merchant adds the SimpleJapanTax component 1258*b* to the registry as follows "SimpleTax.SimpleJapanTax item_included_field item_rate_field" where the "item_included_field" argument specifies a key-value pair which identifies whether taxes are included in the item price, the "item_rate_field" specifies a key-value pair which specifies the tax rate which was charged on the item.

The SimpleJapanTax component 1258*b* then evaluates the ship_to_country key-value pair to determine whether the purchased items are destined for Japan. If so, for every item in the order 124, the SimplefapanTax component 1258*b* applies well known techniques to determine whether taxes are included in the item price by accessing the key-value pair specified in the item_included_field argument. The SimpleJapanTax component 1258*b* then accesses the key-value pair specified by the item_rate_field argument to determine the rate of tax. If the tax is included in the item price, the SimpleJapanTax component 1258*b* stores the amount of included tax in the tax_included key-value pair in each item blackboard 352. The SimpleJapanTax component 1258*b* then sums the tax_included values associated with each item and stores the total included taxes in the order tax_included key-value pair in the order blackboard 350.

If the tax is not included in the item price, the SimpleJapanTax component 1258*b* computes the additional taxes for each item and stores the amount in the tax_total key-value pair in each item blackboard 352. The SimpleJapanTax component 1258*b* then sums the tax_total values associated with each item and stores the total amount in the order tax_total key-value pair in the order blackboard 350.

The SimpleCanadaTax component 1258*c* computes a tax rate for Canada, including a goods and services tax (GST) and a provincial sales tax (PST). GST is common throughout Canada and almost all goods and services have a GST (but not all). If a product is taxed, then the GST rate is the same throughout Canada. However, some products are not taxed and some service merchants (making less than a given amount) may elect to not charge GST. In addition, each province in Canada has its own PST, typically a fixed rate for all products which depends on the product (although a province may elect to tax an item not taxed by another province). Furthermore, when shipping across a province, the consumer does not need pay any PST taxes. Although not common, a product may have PST but not GST (and vice versa).

A merchant adds the SimpleCanadaTax component **1258***c* with the following registry command "SimpleTax.SimpleCanadaTax province . . . " where "province" identifies the province for which to compute the tax. The SimpleCanadaTax component **1258***c* then evaluates the ship_to_country key-value pair to determine whether the purchased items are destined for Canada. If so, the SimpleCanadaTax component **1258***c* applies well-known techniques to calculate the proper Canadian taxes. The SimpleCanadaTax component **1258***c* then stores the amount of taxes for each item in a tax_total key-value pair in each item blackboard **352** and the amount of taxes for the entire order in the order tax_total key-value pair in the order blackboard **350**.

The value added European taxes are calculated with a SimpleVATTax component **1258***d*. A merchant adds the SimpleVATTax component **1258***d* with the following registry command "SimpleTax.SimpleVATTax country rate" where the "country" argument specifies a country name and the "rate" argument specifies a tax rate.

When processing an order **124**, the SimpleVATTax component **1258***d* evaluates the ship_to_country key-value pair to determine whether the purchased items are destined for a specified country in Europe. If so, the SimpleVATTax component **1258***d* applies well-known techniques to calculate the proper value added taxes. The SimpleVATTax component **1258***d* then stores the value added tax for each item in the tax_vat_item key-value pair, the included taxes in the tax_included key-value pairs and the added taxes in the tax_total key-value pairs existing in each item blackboard **352**. The SimpleVATTax component **1258***d* then stores the total amount of the taxes in the tax_included and tax_total key-value pairs in the order blackboard **350**.

The tax required component **1260** generates an error message if the tax_total or tax_included key-value pairs in the order blackboard **350** are not set to a value. The tax required component **1260** stores the error message in the order **124**.

k) The Order Total Stage

The components in the order total stage **384** compute the total charge for the order **124**. The preferred order total default component **1262** sets the order_total key-value pair to the sum of the oadjust_subtotal key-value pair, the shipping_total key-value pair, the tax_total key-value pair, and the handling_total key-value pair.

In the preferred embodiment, neither an order total optional component **1264** nor an order total required component **1266** exist.

l) The Inventory Stage

The components in the inventory stage **386** verify that every selected item is in stock. In the preferred embodiment, a default inventory component **1268** does not exist. However, the preferred embodiment does contain two inventory optional components **1270** called 1) the FlagInventory component **1270***a* and 2) the LocalInventory component **1270***b*.

The FlagInventory component **1270***a* determines whether there is insufficient stock to complete the order **124**. If so, the FlagInventory component **1270***a* indicates the necessity for a back order. As explained above, in the product information stage, a database query based on each item sku determines whether the items are in stock. The results of the database query are stored in the product_in_stock key-value field in the item blackboards **352**. The FlagInventory component **1270***a* checks the product_in_stock key value pair to determine whether the desired items are in stock. If the items are not in stock, the FlagInventory component **1270***a* sets an inventory_backorder key-value pair to indicate that the item is out of stock. The merchant adds the FlagInventory component **1270***a* to the registry with the "WLStdOrder.FlagInventory" command.

The LocalInventory component **1270***b* uses the sku key-value pairs to perform one or more database queries which determine whether the product local inventory amount contains enough of the desired items. If not, the LocalInventory component **1270***b* sets the inventory_backorder flag which indicates to the consumer that the merchant is out of the selected items. The merchant adds the LocalInventory component **1270***b* to the registry with the "WLStdOrder. LocalInventory" command.

In the preferred embodiment, an inventory required component **1272** does not exist.

m) The Payment Stage

The components in the payment stage **388** approve credit-card payments. A payment default component **1274** sets the payment_auth_code key-value pair in the order blackboard **350** to "FAITH." While the preferred embodiment does not have a payment optional component **1276** which performs card authorization, software such as VeriFone's Point of Sale (vPOS) software could be used. VeriFone's Point of Sale (vPOS) software is publicly available and can be obtained from VeriFone, Inc. The payment required component **1278** evaluates whether the value associated with the payment_auth_code key has been set.

n) The Accept Stage

The accept stage **390** generates a completed purchase order. The preferred embodiment does not have an accept default component **1280**. However, the preferred embodiment does contain a number of accept optional components **1282** including: a POGen component **1282***a*, a POGenPipe component **1282***b*, a SaveOrderToDb **1282***c* component and a SaveItemsToDb component **1282***d* and a ReduceLocalInventory component **1282***e*.

The POGen component **1282***a* generates a purchase order by directing the dynamic page generator **120** to generate an HTML page with a purchase order template. The dynamic page generator **120** then stores the HTML purchase order in a file. The name of the file is based on the order identification code in the order_id key-value pair. A merchant adds the POGen component **1282***a* with the following registry command "WLStdOrder.POGen template output-dir" where the "template" argument identifies the purchase order template file and the "output-dir" argument identifies the directory in which to place the created purchase order file.

The POGenPipe component **1282***b* generates a purchase order by directing the dynamic page generator **120** to generate an HTML purchase order with a purchase order template. The POGenPipe **1282***b* component then uses well-known techniques to direct the dynamic page generator **120** to send the HTML purchase order to another program. A merchant adds the POGenPipe component **1282***b* with the following registry command "WLStdOrder.POGenPipe template command" where the "template" argument identifies the purchase order template and the "command" argument identifies the program which receives the product order HTML page.

The SaveOrderToDb component **1282***c* uses well-known database techniques to save a purchase order in a specified database table. A merchant adds the SaveOrderToDb component **1282***c* with the following registry command "WLStdOrder.SaveOrderToDb table translation" where the "table" argument specifies a database table and the "translation" argument is optional and specifies a transaction dictionary which translates the purchase order into a database format.

The SaveItemsToDb component **1282***d* uses well-known database techniques to saves information about the purchased items to a specified database table. A merchant adds the SaveItemsToDb component **1282***d* with the following registry command "WLStdOrder.SaveOrderToDb table translation" where the "table" argument specifies the database table and the "translation" argument is optional and specifies a transaction dictionary which translates the purchase order into a database format. The preferred embodiment does not have an accept required component **1284**.

The ReduceLocalInventory component **1282***e* reduces the inventory in an inventory database **130** by the number of products ordered. The ReduceLocalInventory component **1282***e* uses the sku key-value pairs and the quantity key-value pairs to generate a database query which deduces the quantity amounts from the database **130**. The merchant adds the ReduceLocalInventory component **1282***e* to the registry with the "WLStdOrder.ReduceLocalInventory queryname" command where the "queryname" argument specifies the database query.

IV. Purchase Processing

FIG. **13** illustrates a flow chart of the sequence of states which occur when a consumer accesses the electronic merchandising system **100**. Beginning in a start state **1300**, the present invention proceeds to state **1302** where the consumer directs his consumer browser **110** to access the electronic merchandising system **100**. Proceeding to state **1304**, the consumer views the virtual store displayed by the dynamic page generator **120**.

In state **1304**, the virtual store offers the consumer a number of options. For instance, the consumer can navigate about the virtual store, view different sales departments, obtain information about products offered for sale, select desired items, view a shopping cart of selected items and can purchase selected items. The various options are represented with buttons, menus, or other user interface inputs which contain hyperlinks.

A. Viewing Items

To view detailed information about an item, the consumer selects the image of an item, an item from a menu or items, etc. In the preferred embodiment, each of the item images have a hyperlink which specifies the item's URL. Proceeding to state **1306**, the consumer browser **110** transmits the item's URL to the dynamic page generator **120**.

Proceeding to state **1308**, the preferred embodiment processes an item blackboard **352** to obtain information about the selected item. A detailed flow chart of state **1308** is illustrated in FIG. **14**. In start state **1400**, the dynamic page generator **120** receives the URL which specifies a particular product. Proceeding to state **1402**, the HTML template directs the order manager **322** to create an item blackboard **352** for the selected product. Proceeding to state **1404**, the order manager **322** also adds the product's key-value pairs to the item blackboard **352**. The order manager **322** adds the product's key-value pair to the item blackboard **352**.

While in state **1406**, the order manager **322** passes the item blackboard **352** to the order engine **360**. The order engine **360** then invokes the product information stage **364** and the item price adjust stage **374**. Proceeding to state **1406**, the order engine **360** passes the item blackboard **352** to the product information stage **364**. As discussed above, the preferred product information default component **1200** accesses the sku key-value pair in the item blackboard **352** and performs a database query that retrieves product data corresponding to the sku. The product information default component **1200** then stores the product data into the product key-value pairs existing in the item blackboard **352**. The product information default component **1200** then passes the item blackboard **352** to the components in the item price adjust stage **374**.

Proceeding to state **1408**, the components in the item price adjust state calculate the regular and current prices of an item. The preferred embodiment contains two item price adjust optional components called the SaleAdjust component **1234***a* and the ItemPromo component **1234***b*. The SaleAdjust component **1234***a* accesses the product_sale_start, the product sale_end and the product_sale_price key-value pairs and adjusts the regular and current prices of an item by setting the iadjust_regularprice and the iadjust_currentprice key-value pairs on the item blackboard **352**. The SaleAdjust component **1234***a* then sets the product_currentprice value to the product_sale_price value if the current date is between the sale start date and end date.

The preferred ItemPromo component **1234***b* is further described in a concurrently filed application having the title "Electronic Promotion System For An Electronic Merchant System" which is incorporated herein by reference. The ItemPromo component **1234***b* defines the type of item which receives a discount, the amount of the discount, the date the promotion begins and the date the promotion ends. If a promotion applies, the ItemPromo component **1234***b* stores the sale price in the product_current_price key-value pair.

Proceeding to end state **1410**, the order engine **360** passes the item blackboard **352** back to the dynamic page generator **120**. Proceeding to state **1310** as illustrated in FIG. **13**, the dynamic page generator **120** combines the product information in the item blackboard **352** with the producthtml template to create a product HTML page. The dynamic page generator **120** then sends to the consumer browser **110**. The consumer browser **110** displays the product HTML page and returns to state **1304**.

B. Viewing Subtotals

Retuning to state **1304**, a consumer may desire to periodically view the items the consumer has selected for purchase. To view the selected items, the consumer can select the shopping cart button **412**. In the preferred embodiment, the shopping cart button **412** has a URL which specifies the order.plan action **346**. Proceeding to state **1320**, when the use selects the shopping cart button **412**, the consumer browser **110** transmits the order.plan URL to the action manager 122.

Proceeding to state **1322**, the preferred embodiment processes an order to obtain information and a subtotal of the order. FIG. **15** illustrates the detailed flow chart of the steps performed by a the order.plan action **346** in state **1322**. In start state **1500**, the action manager **122** receives the URL which specifies the orderplan action **346**. Proceeding to state **1502**, the action manager **122** directs the order manager **322** to create an order **124**. Proceeding to state **1504**, the order manager 322 obtains the order information and item information in the order table 304 and stores it as key-value pairs in the order 124.

The action manager 122 then passes the order to the order engine 360. The order engine 360, in turn, passes the order 124 to the components in the product information stage 364, the merchant information stage 366, the shopper information stage 368, the order initialization stage 370, the order check stage 372, the item price adjust stage 374, the order price adjust stage 376, the shipping stage 378, the handling stage 380, the tax stage 382, the order total stage 384 and the inventory stage 386.

Proceeding to state 1506, the order engine 360 passes the order 124 to the product information stage. As discussed above, the components in the product information stage 364 retrieves product data for each item from the products database 130 and writes the product data to the appropriate item blackboard 352. The components in the product information stage 364 then pass the order 124 to the merchant information stage 366.

Proceeding to state 1508, the components in the merchant information stage 366 can obtain merchant information from a merchant database 130. In the preferred embodiment, however, does not contain any merchant information components. Instead, the order 124 is passed to the components in the shopper information stage 368.

Proceeding to state 1510, the components in the shopper information stage 368 set the shopper key-value pairs by retrieving shopper information from a database 130. The preferred shopper information default component 1212 accesses the shopper_id key-value pair in the order blackboard 350 and performs a database query which accesses the identified shopper in the shopper table 300. Each column in the shopper table 300 is then added to a shopper key-value pair in the order blackboard 350. For example, the added shopper key-value pairs can include the shopper_address, the shopper_size, the shopper_preferences, etc. The shopper information default component 1212 then passes the order 124 to the order initialization stage 370.

Proceeding to state 1512, the components in the order initialization stage 370 initialize the order key-value pairs which will be utilized by other components in the order pipeline 362. In particular, the CopyData component 1220*a* copies data from one key-value to another key-value pair. For example, the CopyData component 1220*a* copies the shipping address to the billing address if the client has specified that the items should be shipped to the same location as the consumer's address. The CopyData component 1220*a* then passes the order 124 to the SetData component 1220*b*.

The SetData component 1220*b* then sets the value of key-value pairs. For example, the merchant can direct the SetData component 1220*b* to set the consumer identification code in the order blackboard. The SetData component 1220*a* then passes the order 124 to the components in the order check stage 372.

Proceeding to state 1514, the components in the order check stage 372 verify that the required key-value pairs are present or determine whether certain key-value pairs satisfy desired requirements. One of the order check optional components called the OrderValidate component 1226*a* is configured to check the order blackboard 350 and verify that the required key-value pairs exist or that the required key-value pairs follow other given rules (i.e., is the value in one of the key-value pairs falls within a minimum or maximum value, etc.). Another order check optional component, called the OrderItemValidate component 1226*b* is configured to check for each item, the required item key-value pairs in the item blackboards 352 exist or that the key-value pairs for each item follow other given rules. The preferred order check required component then checks the order 124 to assure that there is at least one item in the order 124. The preferred order check required component then passes the order 124 to the components in the item price adjust stage 374.

Proceeding to state 1516, the components in the item price adjust stage 374 calculate the on-sale price of an item as discussed above and store the discounted value in the iadjust_currentprice key-value pair. The components in the item price adjust stage 374 then pass the order 124 to the components in the order price adjust stage 376.

Proceeding to state 1518, the components in the order price adjust stage 376 set the adjusted price of the entire order 124. The preferred DbOrderPromo component 1240. The DbOrderPromo component 1240 discounts the order amount based on the shopper, the items purchased, the discount award given, etc. The DbOrderPromo component 1240 stores the discounted amount in the oadjust_adjustedprice key-value pair.

The order price adjust required component 1242 completes the order price adjustment by setting, if not already set, the oadjust_adjustedprice key-value pair for each item to the item's current price. In addition, the order price adjust required component 1242 also sets the oadjust_subtotal key-value pair on the order blackboard 350 equal to the sum of all oadjust_adjustedprice key-value pairs associated with each item. The components in the item price adjust stage 374 then pass the order 124 to the components in the shipping stage 378.

Proceeding to state 1520, the components in the shipping stage 378 calculate the total shipping charge. The shipping default component 1244 sets the shipping charges to zero by setting the value in the shipping_total key-value pair existing in the order blackboard 350 to zero. Three optional components exist which a merchant may add to perform shipping calculations: 1) the FixedShipping component 1246*a,* 2) the LinearShipping component 1246*b,* and 3) the TableShipping component 1246*c.*

The FixedShipping component 1246*a,* evaluates the shipping method in the shipping_method key-value pair and charges a fixed shipping amount by setting the shipping_total key-value pair to a fixed amount. The LinearShipping component 1246*b* relies on a rate computed with a basis value (the basis value is the unit of measurement for the product). The basis value is specified by the merchant in the registry. The LinearShipping component 1246*b* multiplies the basis and the rate to determine the shipping cost. The LinearShipping component 1246*b* then stores the shipping amount in the shipping_total key-value pair existing in the order blackboard 350.

The TableShipping component 1246*c* uses a lookup table to determine what the shipping cost should be. The merchant specifies a basis (the unit of measurement for the product), a rate per basis unit, a database query name, and optionally a key-value pair used to calculate the shipping cost. The specified database query searches the database 130 for the proper value. The TableShipping component 1246*c* then stores the shipping cost in the shipping_total key-value pair.

In state 1520, the shipping required component also verifies whether the shipping_total key-value pair contains a value. If not, the shipping required component generates an error message which is stored in the order. The shipping required component then passes the order 124 to the components in the handling stage 380.

Proceeding to state 1522, the components in the handling stage 380 calculate the total handling charge for the order 124. The handling default component 1250 sets the handling charges to zero by setting the handling_total value in the order blackboard 350 to zero. In addition, three optional components exist which a merchant may add to perform handling calculations: 1) the FixedHandling component 1252*a*, 2) the LinearHandling component 1252*b* and 3) the TableHandling component 1252*c*.

The FixedHandling component 1252*a*, evaluates the handling method in the handling_method key-value pair and charges a fixed handling amount by setting the handling_total value to a fixed amount. The merchant specifies a method and corresponding handling price in the registry. The LinearHandling component 1252*b* relies on a rate computed with a basis value that the merchant specifies in the registry and stores the computer handling charges in the handling_total value.

The TableHandling component 1252*c* uses a lookup table to determine what the handling cost should be. The merchant specifies a basis (the unit of measurement for the product), a rate per basis unit, a database query name, and optionally a key-value pair used to calculate the handling cost. The specified database query searches the database 130 for the proper value. The handling cost is then stored in the handling_total key-value pair.

In state 1522, the handling required component 1254 verifies whether the handling_total contains a value. If not, the handling required component 1254 generates an error message. The handling required component 1254 then passes the order 124 to the components in the tax stage 382.

Proceeding to state 1524, the components in the tax stage 382 compute the total tax for a given order 124. The default tax component 1256 sets the tax_total key-value pair in the item blackboards 352 and the tax_total key-value pairs in the order blackboard 350 to zero.

In addition, optional SimpleTax components exist for the tax models used in USA, Canada, Europe, and Japan. The SimpleUSTax component 1258*a* evaluates the ship_to_country key-value pair to determine whether the purchased items are destined for the United States of America. If so, the SimpleUSTax component 1258*a* evaluates the name of the state in the ship_to_state key-value pair and applies the specified tax rate to each item. The SimpleUSTax component 1258*a* then stores for each item and for the entire order, the amount of taxes in the tax_total key-value pair in the order blackboard 350.

The SimpleJapanTax component 1258*b* calculates a tax rate for Japan. The SimpleJapanTax component 1258*b* then evaluates the ship_to_country key-value pair to determine whether the purchased items are destined for Japan. If so, the SimpleJapanTax component 1258*b* stores for each item and for the entire order, the amount of taxes in the tax_total and tax_included key-value pairs.

The SimpleCanadaTax component 1258*c* computes a tax rate for Canada, including a goods and services tax (GST) and a provincial sales tax (PST). The SimpleCanadaTax component 1258*c* evaluates the ship_to_country key-value pair to determine whether the purchased items are destined for Canada. If so, the SimpleCanadaTax component 1258*c* applies well-known techniques to calculate the proper Canadian taxes. The SimpleCanadaTax component 1258*c* then stores for each item and the entire order, the amount of taxes in the tax_total key-value pairs.

The SimpleVATTax component 1258*d* calculates the value added European taxes. When processing the order 124, the SimpleVATTax component 1258*d* evaluates the ship_to_country key-value pair to determine whether the purchased items are destined for a specified country in Europe. If so, the SimpleVATTax component 1258*d* applies well-known techniques to calculate the proper value added taxes. The SimpleVATTax component 1258*d* then stores the amount of taxes for each item and for the entire order in the tax_vat_item, tax_total and the tax_included key-value pairs.

In state 1524, the tax required component 1260 generates an error if the tax_total or tax_included key-value pairs do not contain a value. The tax required component 1260 then passes the order 124 to the components in the order total stage 384.

Proceeding to state 1526, the components in the order total stage 384 computes the total charge for the order 124. The preferred default order total component sets the sets the order_total key-value pair to the sum of the oadjust_subtotal key-value, the shipping_total key-value pair, the tax_total key-value pair, and the handling_total key-value pair. The default order total component then passes the order 124 to the components in the inventory stage 386.

Proceeding to state 1528, the components in the inventory stage 386 verify whether the desired items are in stock. In the preferred embodiment, a default inventory component does not exist. However, the preferred embodiment does contain two inventory optional components called the 1) FlagInventory component 1270*a* and 2) the LocalInventory component 1270*b*.

The FlagInventory component 1270*a* sets values indicating an error if there is insufficient stock and will indicate whether the item is back ordered. In particular, the FlagInventory component 1270*a* checks the product_in_stock key-value field in the item blackboards 352 to determine whether an item is in stock. The product_in_stock key-value pair is obtained by using the sku key-value pair to generate a database query which determines whether the identified item is in stock. If the item is not in inventory, the FlagInventory component 1270*a* sets an inventory_backorder key-value pair to indicate that the item is out of stock.

When the LocalInventory component 1270*b* receives the order 124, the LocalInventory component 1270*b* checks for inventory in a local database 130 and will ensure that the product_local_inventory key-value pair is at least as large as the total number of items in inventory. The LocalInventory component 1270*b* uses the sku key-value pairs to perform one or more database queries which determine whether the product local inventory amount contains enough of the desire items. If not, the LocalInventory component 1270*b* sets the inventory_backorder flag which indicates to the shopper that the merchant is out of the selected items.

Proceeding to end state 1530, the order engine 360 passes the order 124 back to the order manager 322. Proceeding to state 1324 in FIG. 13, the order manager 322 retrieves the item key-value pairs and stores them in the order table 304. In state 1324, the dynamic page generator 120 then creates the shopping cart HTML page 400 is illustrated in FIG. 4. In particular, the dynamic page generator 120 uses the item's stock keeping unit, description, color, size, list price, sale price, quantity, extra discounts and total price in the key-value pairs to create the shopping cart HTML page 400. The consumer browser 110 then displays the shopping cart HTML page 400.

C. Completing A Sales Transaction

Returning to state 1304, to a consumer may decide to initiate a sales transaction. To initiate a sales transaction, the consumer selects the check out button 414. In the preferred embodiment, the check out button 414 has a hyperlink which specifies a URL which invokes the order.plan action 346 in the action manager 122. Proceeding to state 1330, the consumer selects the check out button 414 and the consumer browser 110 transmits the URL for the order.plan action 346 to the action manager 122.

Proceeding to state 1332, the order.plan action 346 as discussed above with respect to FIG. 15, directs the order manager 322 to create an order 124. The order.plan action 346 then passes the order 124 to the order engine 360 which directs the order 124 to the components in the product information stage 364, the merchant information stage 366, the shopper information stage 368, the order initialization stage 370, the order check stage 372, the item price adjust stage 374, the order price adjust stage 376, the shipping stag, the handling stage 380, the tax stage 382, the order total stage 384 and the inventory stage 386.

Proceeding to state 1334, the order.plan action directs the page generator to create the acceptance check out HTML page 600 illustrated in FIG. 6. The acceptance check out HTML page 600 which displays the total price of the selected items and which also requests the billing information as illustrated in FIG. 15. Proceeding to state 1336, the consumer can enter her or his credit card information and select the purchase now button. The purchase now button contains a URL which invokes the order.purchase action 348 in the action manager 122.

Proceeding to state 1338, the consumer selects the purchase now button and the consumer browser 110 sends the order.purchase action URL to the action manager 122. FIG. 16 illustrates a detailed block diagram of the states in the order.purchase action 348. Begining in a start state 1600, the order.purchase action 348 proceeds to state 1602 and directs the order manager 322 to create an order 124. Proceeding to state 1604, the order manager adds the order key-value pairs to the order blackboard 350 and the item key-value pairs to the item blackboard 352. The order.purchase action 348 then passes the order 124 to the order engine 360. The order engine 360 then passes the order to each in the stages in the order pipeline 362 including the payment stage 388 and the accept stage 390.

In states 1606–1638 order engine 360 order pipeline passes the order to the components in the product information stage 364, the merchant information stage 366, the shopper information stage 368, the order initialization stage 370, the order check stage 372, the item price adjust stage 374, the order price adjust stage 376, the shipping stage 378, the handling stage 380, the tax stage 382, the order total stage 384, and the inventory stage 386. The processing of the order with respect to these stages is discussed above. In state 1628, the components in the inventory stage 386 then passes the order 124 to the components in the payment stage 388.

Proceeding to state 1630, the components in the payment stage 388 then approve the credit-card payments. In the preferred embodiment, the payment default component 1274 sets the payment_auth_code key-value pair in the order blackboard 350 to "FAITH." While the preferred embodiment does not have a payment optional component 1276 which performs card authorization, software such as VeriFone's Point of Sale (vPOS) software could be used. VeriFone's Point of Sale (VPOS) software is publicly available and can be obtained from VeriFone, Inc.

The payment default component 1274 then passes the order 124 to the payment required component 1278 which evaluates the payment_auth_code key-value pair to determine whether the value associated with the payment_auth_code key has been set. The payment required component 1278 then passes the order 124 to the components in the accept stage 390.

Proceeding to state 1632, the components in the accept stage 390 generate a completed purchase order. The preferred embodiment contains a number of accept optional components including: the POGen component 1282*a*, the POGenPipe component 1282*b*, the SaveOrderToDb component 1282*c* and the SaveItemsToDb component 1282*d*. The POGen component 1282*a* generates a purchase order by directing the dynamic page generator 120 to generate an HTML page with a purchase order HTML template.

The POGenPipe component 1282*b* generates a purchase order by directing the dynamic page generator 120 to generate an HTML purchase order with a purchase order HTML template. The POGenPipe component 1282*b* then uses standard techniques such as named pipes to direct the dynamic page generator 120 to send the HTML purchase order to another program. The SaveOrderToDb component 1282*c* uses well-known database techniques to save a purchase order in a specified database 130. The SaveItemsToDb component 1282*d* uses well-known database techniques to saves information about the purchased items to a specified database 130.

When the ReduceLocalInventory component 1282*e* receives the order 124, the ReduceLocalInventory component 1282*e* reduces the inventory in an inventory database 130 by the number of products ordered. The ReduceLocalInventory component 1282*e* uses the sku key-value pairs and the quantity key-value pairs to specify a database query which deducts the quantity amounts from the database 130.

Proceeding to end state 1634, the preferred embodiment proceeds to state 1340 in FIG. 13 and sends the purchased items to the consumer. Returning to state 1304, the consumer can exit the electronic merchandising system 110 by navigating to another website or by exiting his consumer browser in end state 1350.

V. Conclusion

While certain preferred embodiments of the invention have been described, these embodiments have been presented by way of example only, and are not intended to limit the scope of the present invention. For example, although described herein with reference to a set of components in the order pipeline 362, a merchant can add other components to customize the flexible electronic merchandising system 100 for the merchant's unique sales transactions. Accordingly, the breadth and scope of the present invention should be defined only in accordance with the following claims and their equivalents.

What is claimed is:

1. A merchandising system for processing electronic sales transactions, said merchandising system comprising:

a computer processor;

an order stored in a computer readable storage media, said order comprising a set of key-value pairs such that said key-value pairs contain information about said sales transaction; and

an order processing module stored in a computer readable storage media, said order processing module executable in said computer processor, said order processing module configured to receive said order and process said key-value pairs in said order;

wherein said order processing module further comprises at least one order initialization component which is configured to initialize at least one of said key-value pairs.

2. The merchandising system of claim 1 wherein said order further comprises multiple sets of said key-value pairs such that one or more said sets of key-value pairs corresponds to an item in said sales transaction.

3. The merchandising system of claim 2 wherein said order processing module further comprises at least one product information component which is configured to obtain information about at least one of said items in said sales transaction and store said information in at least one of said key-value pairs.

4. The merchandising system of claim 2 wherein said order processing module further comprises at least one item adjust component which is configured to determine a discount price associated with one of said items in said order and store said discount price in at least one of said key-value pairs.

5. The merchandising system of claim 1 wherein said key-value pairs in said order are configured to store information for different types of sales transactions.

6. The merchandising system of claim 5 wherein said order processing module further comprises different components which process one or more of said key-value pairs.

7. The merchandising system of claim 6 wherein said components are configured to utilize one or more of said key-value pairs to perform a specified function.

8. The merchandising system of claim 1 wherein said order processing module further comprises at least one merchant information component which is configured to obtain merchant information and store said merchant information in at least one of said key-value pairs.

9. The merchandising system of claim 1 wherein said order processing module further comprises at least one order check component which is configured to ensure that at least one of said key-value pairs exists in said order.

10. The merchandising system of claim 1 wherein said order processing module further comprises at least one order price adjust component which is configured to determine a discount price associated with said order and store said discount price in at least one of said key-value pairs.

11. The merchandising system of claim 1 wherein said order processing module further comprises at least one shipping component which is configured to determine a shipping cost associated with said order and store said shipping cost in at least one of said key-value pairs.

12. The merchandising system of claim 1 wherein said order processing module further comprises at least one handling component which is configured to determine a handling cost associated with said order and store said handling cost in at least one of said key-value pairs.

13. The merchandising system of claim 1 wherein said order processing module further comprises at least one tax component which is configured to determine an amount of taxes associated with said order and store said amount of taxes in at least one of said key-value pairs.

14. The merchandising system of claim 1 wherein said order processing module further comprises at least one order total component which is configured to determine a price amount associated with said order and store said price amount in at least one of said key-value pairs.

15. The merchandising system of claim 1 wherein said order processing module further comprises at least one inventory component which is configured to determine whether sufficient inventory exists to fulfill said order.

16. The merchandising system of claim 1 wherein said order processing module further comprises at least one payment component which is configured to obtain an authorization to bill an amount associated with said order to an account, said payment component further configured to store said authorization in one of said key-value pairs.

17. The merchandising system of claim 1 wherein said order processing module further comprises at least one accept component which is configured to generate a purchase order based on said key-value pairs in said order.

18. A merchandising system for processing electronic sales transactions, said merchandising system comprising:

a computer processor;

an order stored in a computer readable storage media, said order comprising a set of key-value pairs such that said key-value pairs contain information about said sales transaction; and

an order processing module stored in a computer readable storage media, said order processing module executable in said computer processor, said order processing module configured to receive said order and process said key-value pairs in said order;

wherein said order processing module further comprises at least one consumer information component which is configured to obtain information about and store said consumer information in at least one of said key-value pairs.

19. A system for processing requests for information about goods or services, said system comprising:

a computer network wherein a plurality of computers have access to said computer network;

an action manager module executing in said computer network, said action manager configured to receive requests for information about goods or services from any one of said plurality of computers, said action manager further configured to create an information object which contains a set of key-value pairs such that said key-value pairs store data about said request; and

an information processing module in communication with said information object such that said information processing module is configured to obtain said data about said goods or services and store said data in at least one of said key-value pairs;

wherein said request for information about said goods or services is associated with a sales transaction;

wherein said information object is an order which contains key-value pairs associated with said sales transaction;

wherein said action manager further comprises an order plan action which is configured to receive said requests;

wherein said order plan action is further configured to add key-value pairs to said order for each of said goods or services in said request and to pass said order to said order processing module.

20. The system of claim 19 where said network of computers is the internet.

21. The system of claim 19 wherein the structure of said information object contains multiple sets of said key value pairs such that each set of key-value pairs corresponds to each good or service identified in said request.

22. The system of claim 19 wherein said request for information about said goods or services is associated with a sales transaction.

23. The system of claim 22 wherein said information object is an order which contains key-value pairs associated with said sales transaction.

24. The system of claim 23 wherein said system further comprises an order table which stores information about said order and copies of said key-value pairs existing in said order.

25. The system of claim 23 wherein said action manager further comprises an add item action which is configured to receive said requests.

26. The system of claim 25 wherein said add item action is further configured to add key-value pairs to said order based on said information about one of said goods or services identified in said request and to pass said order to said order processing module.

27. The system of claim 26 wherein said order processing module is further configured process said order to obtain data about said one of said goods or services in said order, and wherein said order processing module is further configured to store said data in at least one of said key-value pairs.

28. The system of claim 23 wherein said order processing module is further configured to process said order to obtain a price amount associated with said sales transaction.

29. The system of claim 23 wherein said action manager further comprises a purchase action which is configured to receive said requests.

30. The system of claim 23 wherein said order processing module is further configured to respond to said purchase action and process said order to perform said sales transaction.

31. A system for processing requests for information about goods or services, said system comprising:

a computer network wherein a plurality of computers have access to said computer network;

an action manager module executing in said computer network, said action manager configured to receive requests for information about goods or services from any one of said plurality of computers, said action manager further configured to create an information object which contains a set of key-value pairs such that said key-value pairs store data about said request; and

an information processing module in communication with said information object such that said information processing module is configured to obtain said data about said goods or services and store said data in at least one of said key-value pairs;

wherein said action manager further comprises a purchase action which is configured to receive said requests;

wherein said purchase action is further configured to add key-value pairs associated with said request to said order and to pass said order to said order processing module.

32. A system for processing requests for information about goods or services, said system comprising:

a consumer means for generating a request for information;

an order means for storing a set of key-value pairs which contain data about said request; and

an action manager means for receiving said request and for adding said key-value pairs which contain data about said request to said order means.

33. The system of claim 32 wherein said request for information identifies a plurality of items.

34. The system of claim 33 wherein said order contains multiple sets of key-value pairs such that at each of said sets of said key-value pairs corresponds to one of said items.

35. The system of claim 34 wherein said order processing means further comprises a product information means for obtaining said information and for storing said information in at least one of said key-value pairs.

36. The system of claim 34 wherein said order processing means further comprises a merchant information means for obtaining merchant data about a merchant associated with said order, and for storing said merchant information in at least one of said key-value pairs.

37. The system of claim 34 wherein said order processing means further comprises a shopper information means for obtaining information about a shopper which initiates said request and for storing said shopper information in at least one of said key-value pairs.

38. The system of claim 34 wherein said order processing means further comprises an order initialization means for setting at least one of said key-value pairs to an initial value.

39. The system of claim 34 wherein said order processing means further comprises an order check means for ensuring at least one of said key-value pairs exist in said order.

40. The system of claim 34 wherein said order processing means further comprises an item price adjust means for determining a discount price associated with one of said items in said order and for storing said discount price in at least one of said key-value pairs.

41. The system of claim 34 wherein said order processing means further comprises an order price adjust means for determining a discount price associated with said order and for storing said discount price in at least one of said key-value pairs.

42. The system of claim 34 wherein said order processing means further comprises a shipping means for determining a shipping price associated with said order and for storing said shipping price in at least one of said key-value pairs.

43. The system of claim 34 wherein said order processing means further comprises a handling means for determining a handling price associated with said order and for storing said handling price in at least one of said key-value pairs.

44. The system of claim 34 wherein said order processing means further comprises a tax means for determining the amount of taxes associated with said order and for storing said taxes in at least one of said key-value pairs.

45. The system of claim 34 wherein said order processing means further comprises an order total means for determining an amount associated with said order and for storing said amount in at least one of said key-value pairs.

46. The system of claim 34 wherein said order processing means further comprises an inventory means for determining the amount of inventory needed to fulfill said order.

47. The system of claim 34 wherein said order processing means further comprises a payment means for obtaining an authorization to bill an amount associated with said order to an account, said payment component means also for storing said authorization in one of said key-value pairs.

48. The system of claim 34 wherein said order processing means further comprises an accept means for generating a purchase order for said order.

49. The system of claim 32 further comprising an information processing means for processing said order means to obtain said information requested by said request, said order information processing means also for storing said information in at least one of said key-value pairs.

50. A merchandising system for processing electronic sales transactions, said merchandising system comprising:

an order comprising a set of order values indicating information about said sales transaction;

a plurality of order processing components configured to receive and process the order in turn;

individual order processing components being configured to receive the order with existing order values, to examine the existing order values, and to potentially add order values or modify the order values depending on the existing order values.

51. An order processing system as recited in claim 50, wherein the order processing components are customizable by a merchant.

52. An order processing system as recited in claim 50, wherein individual components can be replaced by a merchant to meet needs that are particular to that merchant.

53. An order processing system as recited in claim 50, wherein individual components can be added to the plurality of individual components by a merchant to meet needs that are particular to that merchant.

54. An order processing system as recited in claim 50, wherein the order contains key-value pairs, the key-value pairs containing the order values.

55. An order processing system as recited in claim 50, wherein the order comprises an order blackboard and one or more item blackboards;

the order blackboard having a plurality of key-value pairs containing information about the order;

each item blackboard having a plurality of key-value pairs containing information about an individual item of the order.

56. An order processing system as recited in claim 50, wherein:

the order values are indexed by corresponding keys;

each order processing component processes a subset of the order values;

each order processing component identifies its subset of order values by the keys of the order values.

57. An order processing system as recited in claim 50, wherein communication between the order processing components is by way of the order values.

58. An order processing system as recited in claim 50, further comprising an order engine that passes the order to the respective order processing components in turn.

* * * * *

US005960408A

United States Patent [19]

Martin et al.

[11] **Patent Number: 5,960,408**

[45] **Date of Patent: *Sep. 28, 1999**

[54] **ON-TIME DELIVERY, TRACKING AND REPORTING**

[75] Inventors: **Doreen J. Martin**; **Guy M. Givens**, both of Boise; **Justin D. Kuttler**, Meridian, all of Id.

[73] Assignee: **Micron Technology, Inc.**, Boise, Id.

[*] Notice: This patent is subject to a terminal disclaimer.

[21] Appl. No.: **09/137,651**

[22] Filed: **Aug. 20, 1998**

Related U.S. Application Data

[63] Continuation of application No. 08/794,155, Feb. 3, 1997, Pat. No. 5,809,479, which is a continuation of application No. 08/278,183, Jul. 21, 1994, abandoned.

[51] **Int. Cl.**^{6} **G06F 17/60**

[52] **U.S. Cl.** **705/11**; 235/376; 235/385; 705/28; 705/29

[58] **Field of Search** 705/1, 11, 22, 705/24, 28, 29; 235/375, 376, 377, 385

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,887,208	12/1989	Schneider et al.	705/28
5,038,283	8/1991	Caveney	705/28
5,117,096	5/1992	Bauer et al.	235/375
5,168,445	12/1992	Kawashima et al.	705/10
5,465,291	11/1995	Barrus et al.	379/67.1

OTHER PUBLICATIONS

"The Service Edge—For Creating and Maintaining Distinctive Customer Service," Lakewood Publications, Mar. 1993, vol. 6, No. 3, pp. 1–3.

Letter dated Oct. 27, 1992 to selected customers.

Dialog Abstract: File 751, Acc. #00244194; Visual/XL Purchasing/Inventory; Smith, Dennis & Gaylord, Inc.; 1981.

Dialog Abstract: File 6, Acc #1708417; Potvin et al.; "Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows"; Montreal Univ. (Quebec); c1993; 22p.

Dialog: File 148, Acc #03882150; Zurier; "Delivering Quality Customer Service"; *Industrial Distribution*; v78 n3; p30(4); Mar. 1989.

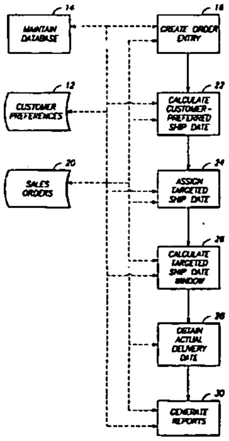
(List continued on next page.)

Primary Examiner—Stephen R. Tkacs
Attorney, Agent, or Firm—Wells, St. John, Roberts, Gregory & Matkin, P.S.

[57] **ABSTRACT**

A computer system is programmed for setting and reporting product delivery dates. The invention includes a step of maintaining a customer preferences database having delivery and reporting preferences for individual customers. The preferences include preferred early and late delivery limits, preferred performance measurement species, and desired advance delivery times. The invention further includes a step of creating a customer order entry for a particular customer. The customer order entry includes a customer-requested delivery date supplied by the customer. A customer-preferred ship date is calculated for the customer order entry based at least in part upon the customer-requested delivery date and at least in part upon the particular customer's specified desired advance delivery time. The customer order entry is then routed to an order scheduler. The computer system shows the order scheduler the calculated customer-preferred ship date and obtains a targeted ship date for the customer order entry from the order scheduler. The system is programmed to then calculate a targeted ship date window based upon the targeted ship date and the particular customer's preferred early and late delivery limits. On-time product delivery statistics are generated for individual customers in accordance with the individual customers' specified preferred performance measurement species.

22 Claims, 1 Drawing Sheet


14
MAINTAIN DATABASE
16
CREATE ORDER ENTRY
12
CUSTOMER PREFERENCES
22
CALCULATE CUSTOMER-PREFERRED SHIP DATE
20
SALES ORDERS
24
ASSIGN TARGETED SHIP DATE
26
CALCULATE TARGETED SHIP DATE WINDOW
28
OBTAIN ACTUAL DELIVERY DATE
30
GENERATE REPORTS

OTHER PUBLICATIONS

Dialog: File 148, Acc #04838628; Raia; "JIT Delivery: Redefining 'On-time'"; *Purchasing*; v109 n3; p64(7); Sep. 13, 1990.

Dialog Abstract: File 15, Acc #00522133; Bowman; "Using 3^{rd} Parties for International Success: Target Stores"; *Distribution*; v89 n10; pp. 78, 80; Oct. 1990.

Dialog: File 148, Acc #05578823; Richardson; "T&D Recognizes Excellence in Logistics"; *Transportation & Distribution*; v32 n12; p27(5); Dec. 1991.

Dialog: File 16, Acc #04289714; Miller; "Don't Let 'Undependable' Suppliers Bug You: Five Principles to Help Boost the Rate of On-time Deliveries"; *Electronic Buyer's News*; Feb. 1, 1993; p. 14.

Dialog: File 16, Acc #04613164; Miller; The Key to Recognition is Teamwork; *Electronic Buyers's News*; Aug. 2, 1993; p. 5.

Dialog Abstract: File 751, Acc #00258029; Macola Purchase Order and Receiving 6.0; Macola Software; First Installed: 1993.

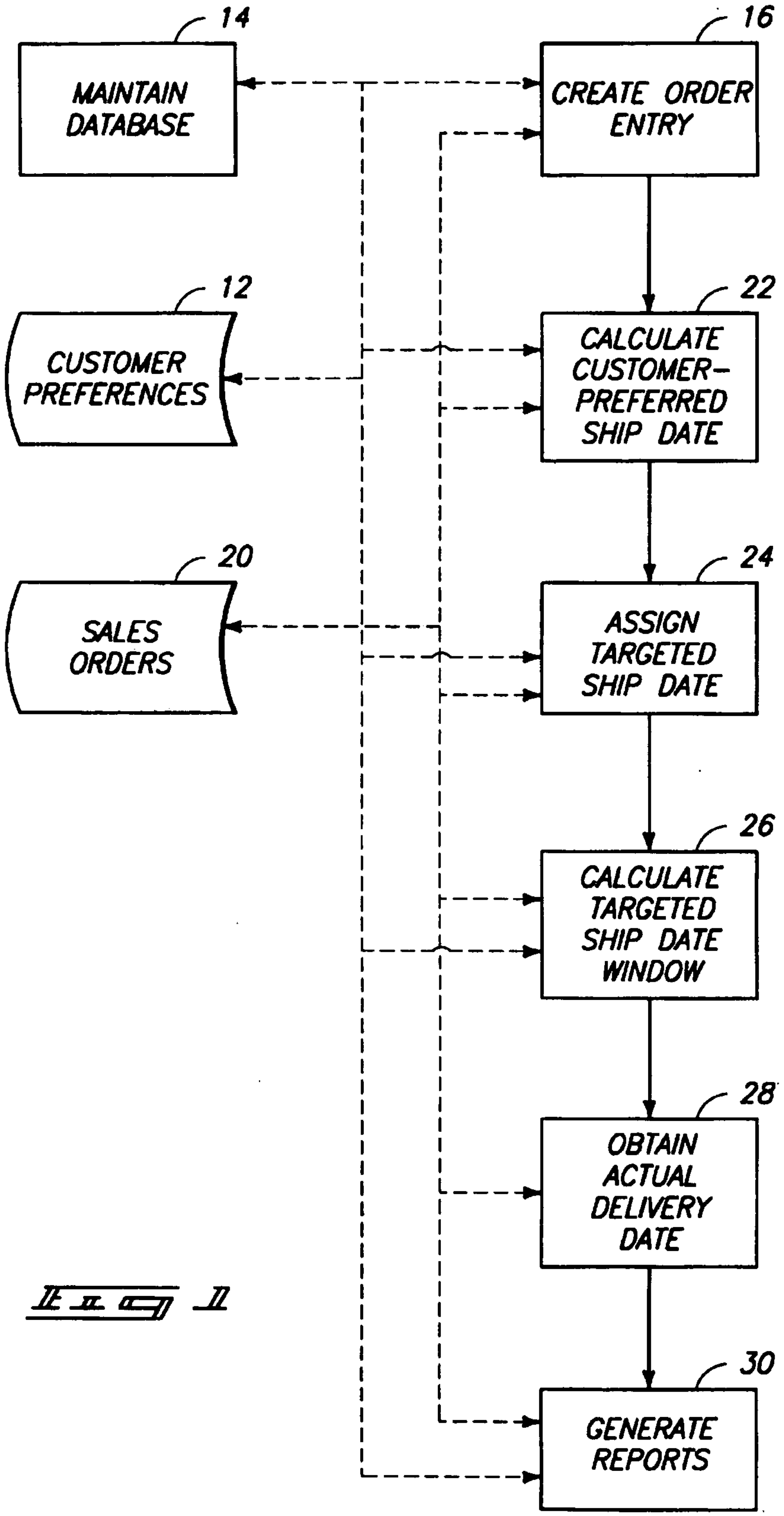

14
MAINTAIN DATABASE
16
CREATE ORDER ENTRY
12
CUSTOMER PREFERENCES
22
CALCULATE CUSTOMER-PREFERRED SHIP DATE
20
SALES ORDERS
24
ASSIGN TARGETED SHIP DATE
26
CALCULATE TARGETED SHIP DATE WINDOW
28
OBTAIN ACTUAL DELIVERY DATE
30
GENERATE REPORTS

Fig 1

ON-TIME DELIVERY, TRACKING AND REPORTING

CROSS REFERENCE TO RELATED APPLICATION

This is a Continuation of U.S. patent application Ser. No. 08/794,155, filed Feb. 3, 1997, and titled "On-Time Delivery, Tracking and Reporting", now U.S. Pat. No. 5,809,479, which in turn is a File Wrapper Continuation of Ser. No. 08/278,183, filed Jul. 21, 1994, now abandoned.

TECHNICAL FIELD

This invention relates to customer order entry and delivery tracking systems.

BACKGROUND OF THE INVENTION

Service quality has become almost as important to many customers as the quality of supplied goods. More and more, customers tend to make purchasing decisions based upon service performance of suppliers. Thus, it has become very important for suppliers to evaluate their own service performance and to provide the results of such evaluations to customers. Suppliers and their customers often cooperate to evaluate service performance.

On-time product delivery is an important service component of many high-volume supply businesses. A customer typically orders a product for delivery on a specified date. The customer expects that the delivery will be no later than that date. However, the customer also does not want the delivery to be too early. The customer considers the delivery to be on time only if it is within an on-time window.

Each customer typically uses its own criteria to calculate the on-time window. For instance, one customer might consider a delivery to be on time if it is no earlier than five days and no later than one day from the requested delivery date. Another customer might consider a delivery to be on time if it is no earlier than four days and no later than the requested delivery date. The fact that each customer uses its own criteria makes it difficult for a supplier to evaluate its own performance. In fact, there are several other variable criteria which make the task of evaluation even more difficult for the supplier.

For instance, different customers use different performance measurement species or calculation units in calculating delivery performance. One customer might use dollar amounts as the measurement species. In such a case, an on-time evaluation would compare the dollar value of on-time items with the dollar value of items which were not on time. Another customer might use product units as a measurement basis. The on-time evaluation would indicate the number of product units delivered on time compared to the number of product units delivered too early or late. A third customer might use order line items or discrete shipments as a statistical calculation basis. The on-time evaluation would indicate the number of on-time shipments compared to the number of other shipments. Variations or averages based on the above schemes might also be used. Furthermore, some customers allow partial on-time credit for partial shipments, while other customers consider a particular order to be late if any part of the order is delivered late.

As another example of varying customer evaluation criteria, some customers equate a "delivery date" with an actual ship date. When such customers request a particular delivery date, they are referring to the day the shipment leaves the supplier's dock. Other customers equate the "delivery date" with the dock date—the date the shipment actually arrives at the customer's dock.

As a further specific example, certain companies might include weekends and holidays in their early and late calculations, while other companies exclude weekends and holidays. Other variable on-time evaluation criteria, not discussed, can of course be used by individual companies.

The highly variable nature of the criteria used to evaluate supplier performance and to compile on-time delivery statistics makes it very difficult for a supplier to either perform to the customers' varying expectations or to even evaluate whether performance expectations are being met. The invention described below addresses this problem. The invention helps a supplier set targeted delivery dates and goals within each customer's expectations, while also providing a statistical analysis of on-time deliveries in accordance with each customer's own evaluation preferences.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a computer flow chart and system diagram which illustrates a preferred embodiment of the invention. Solid lines indicate process flow. Dashed lines indicate data transfer between specific processes and physical data storage devices.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

This disclosure of the invention is submitted in furtherance of the constitutional purposes of the U.S. Patent Laws "to promote the progress of science and useful arts." U.S. Constitution, Article 1, Section 8.

The invention is an on-time delivery tracking and reporting computer system which maintains customer order and delivery information. The computer system implements a method of both setting and reporting product delivery dates in accordance with individual customers' expectations. The computer system includes a conventional data processor which is programmed using conventional programming techniques to perform the functions described below. The programmed data processor thus forms means for accomplishing the described functions.

FIG. 1 is a flow chart and system diagram showing a preferred embodiment of the invention. The invention includes a customer preferences database **12** having delivery and reporting preferences for individual customers of a particular supplier. The invention also includes an independent step **14** of maintaining customer preferences database **12**. Database maintenance is typically performed by customer service representatives of the supplier, from within the supplier's order processing computer system.

The delivery and reporting preferences contained within database **12** include preferred early and late delivery limits, preferred performance measurement species, and desired advance delivery times. The early and late delivery limits specify, for individual customers, on-time windows relative to delivery dates which are requested or expected by the individual customers. The preferred performance measurement species indicate, for individual customers, the customer-preferred bases for statistical measurements—whether the customers measure performance in terms of dollars, units, line items, or shipments. The desired advance delivery times indicate, for individual customers, the number of days by which the customers prefer their orders to be early. For instance, a particular customer might specify early and late delivery limits of five and zero days respectively. However, the customer might prefer a delivery date of two days early. The desired advance delivery time indicates this preference.

Customer preferences database **12** further includes ship/dock flags for individual customers to indicate whether said customers consider delivery dates for particular products to be ship dates (the dates the products leave the suppliers' sites) or dock dates (the dates the products arrive at the customers' sites). Days-in-week variables in the delivery and reporting preferences specify the number of days counted in each week by particular customers. These variables are generally used to indicate whether particular suppliers count weekends in their on-time calculations.

The delivery and reporting preferences of database **12** also include partial shipment allowed flags for respective customers, indicating whether the customers give credit for on time delivery of partial shipments. The delivery and reporting preferences further include customer-preferred reporting periods, indicating the preferred reporting intervals for individual customers.

Other customer preferences might also be included in database **12**, indicating such information as whether customers will allow rescheduling of shipments, calendar holidays for each customer, and/or colendar holidays for the supplier. Customer preferences database **12** will preferably be updated at least once every year for each customer, or as otherwise determined to be needed. The system is programmed to prompt customer service representatives whenever updating is needed, based upon programmed updating periods which are initially specified by the customer service representatives.

The preferred embodiment of the invention includes a step **16** of creating a customer order entry for a particular customer. The computer system is programmed to reference customer preferences database **12** during the order entry process to set preferable delivery dates for individual customers. These dates are based upon customer-requested delivery dates supplied by the customers. Individual customer order entries are created and stored in a sales orders database **20**.

The invention includes a step **22** of calculating a customer-preferred ship date for the customer order entry based at least in part upon the customer-requested delivery date and at least in part upon the particular customer's specified preferences as maintained in customer preferences database **12**. More specifically, step **22** includes a step of subtracting the customer's desired advance delivery time from the customer-requested delivery date to arrive at the customer-preferred ship date. Step **22** also includes checking the customer's ship/dock flag, and then pre-dating the customer-preferred ship date if the customer's ship/dock flag indicates that the particular customer considers a delivery date to be a dock date rather than a ship date. The customer-preferred ship date would be pre-dated in this case by the expected shipping time requirements to the customer's site. The resulting customer-preferred ship date would represent the date the customer order would have to leave the supplier's site to arrive at the customer's site on the date preferred by the customer.

The information entered so far relates only to the customers' requested delivery dates. In some cases, it might not be possible for the supplier to meet a requested or preferred date. Accordingly, the customer order entry is routed to a human order scheduler for assignment of a targeted ship date, as indicated in FIG. 1 by block **24**. Based upon the information contained in customer preferences database **12** and sales orders database **20**, the computer system is programmed to show the order scheduler the calculated customer-preferred ship date and to obtain from the scheduler a targeted ship date for the customer order entry. This date may or may not correspond to the customer-preferred ship date. Regardless, all dates are recalculated based on the targeted ship date, and a customer-expected delivery date is established in sales orders database **20** and communicated to the customer. The customer-expected delivery date accounts for customer preferences variables as discussed above. More specifically, the customer-expected delivery date is calculated by adding a shipping delay to the targeted ship date, depending on the status of the customer's ship/dock flag, and by also adding the customer's desired advance delivery time to the targeted ship date.

In a further step **26**, the computer system is programmed to calculate a targeted ship date window based upon the targeted ship date and the particular customer's preferred early and late delivery limits. The targeted ship date window is obtained by simply subtracting and adding, respectively, the early and late delivery limits from the targeted ship date. The targeted ship date window gives the range of actual ship dates which will result in an on time delivery to the customer, based upon the customer's own rules. The sales order entries, including targeted ship dates, are thus completed with information obtained both from each order entry and from information stored on a long-term basis in customer preferences database **12**.

As an example of the order entry process, suppose a customer requests delivery of a shipment on March 17. The customer preferences database for that customer indicates early and late delivery limits of four and zero days, respectively. Delivery dates are specified in terms of dock dates, as indicated by the customer's ship/dock flag. The customer's desired advance delivery time indicates that the customer prefers deliveries to be two days early. Order entry personnel select 2nd day air delivery.

The order is routed to a scheduler who is shown a customer-preferred shipment date of March 13, which accounts for two day delivery and the customer's desired advance delivery time of two days. The scheduler realizes that the shipment cannot be made on that date, and enters a targeted ship date of March 22. A customer-expected delivery date of March 26 is then calculated by adding the delivery delay and the customer's desired advance delivery time to the targeted ship date. The customer-expected delivery date is communicated to the customer, which then uses this date for purposes of on-time measurements.

Once delivery has taken place, the computer system is programmed in a step **28** to obtain actual delivery dates for each shipment. If ship dates are being used as delivery dates by a particular customer, ship dates as obtained from the supplier's own records are used as delivery dates. If customer dock dates are being used instead as delivery dates, such dock dates can be obtained either from the customer or from the carrier. Electronic data interchange (EDI) can be used in either case to obtain the necessary data. The actual delivery dates are stored in sales orders database **20**.

After obtaining delivery dates, the computer system in accordance with the invention is programmed in a step **30** to generate on-time product delivery statistics for individual customers. The determination of whether a particular product delivery is on time is based upon the customer-expected delivery date and upon the customer's preferred early and late delivery limits. Step **30** includes generating statistical reports for each customer. The reports are generated periodically, for each customer, based upon that customer's specified reporting periods. For instance, some customers might require weekly reports, while other customers desire only monthly reports.

Each report is formatted in accordance with an individual customer's specified preferred performance measurement species. For instance, if customer preferences database **12** indicates that a particular customer measures performance in terms of dollars, the report for that customer might appear as follows:

Total Dollar Shipments	$1,000,000
On-time Shipments	$700,000
Percentage On-Time	70%

However, if the same customer measures performance in terms of actual units, the report for that customer might appear as follows:

Total Units Shipped	300,000
On-time Units	200,000
Percentage On-Time	67%

If the customer measures performance in terms of shipments or line items, the report might appear as follows:

Total Shipments	10
On-time Shipments	8
Percentage On-Time	80%

The determination of whether a particular product delivery is on time is based upon the supplier's targeted ship date window when the customer's ship/dock flag indicates that that customer equates delivery dates with actual ship dates. When delivery dates specify dock dates, the determination is based upon a window calculated from the customer-expected delivery date and upon the early and late delivery limits. In either case, the determination is also based upon the customer's days-in-week variable. Furthermore, in compiling the reports, the computer system is programmed to count a partial delivery which is on time as a fraction of an on-time delivery if a customer's partial shipment allowed flag indicates that the customer allows partial on-time shipments to be counted in on-time statistics.

Once deliveries have been completed, it is also desirable to determine and document the reasons for any deliveries which were not on time. Accordingly, provisions are provided for recording reasons for late shipments or for shipments which have to be rescheduled.

The system and program described above allow a supplier to easily measure its performance using the same evaluation criteria used by its customers. This not only helps the supplier perform to the customer's expectations, but provides an additional service to customers in the form of on-time reports in the formats needed and actually used by the customers. Furthermore, the system allows the supplier to take advantage of customers' delivery windows. Rigid internal rules, which did not account for individual customers, had previously prevented this. It is anticipated that the system will increase supplier performance, particularly in allowing the supplier to provide a higher percentage of on-time deliveries.

In compliance with the statute, the invention has been described in language more or less specific as to structural and methodical features. It is to be understood, however, that the invention is not limited to the specific features shown and described, since the means herein disclosed comprise preferred forms of putting the invention into effect. The invention is, therefore, claimed in any of its forms or modifications within the proper scope of the appended claims appropriately interpreted in accordance with the doctrine of equivalents.

We claim:

1. A computer system for maintaining customer order and delivery information, the computer system being programmed to:

maintain a customer preferences database having delivery and reporting preferences for individual customers, said preferences including preferred early and late delivery limits;

store ship/dock flags in the delivery and reporting preferences for individual customers to indicate whether said customers consider delivery dates for particular products to be ship dates or dock dates;

store desired advance delivery times in the delivery and reporting preferences for individual customers;

create a customer order entry for a particular customer, the customer order entry including a customer-requested delivery date supplied by the particular customer;

calculate a customer-preferred ship date for the customer order entry based at least in part upon the customer-requested delivery date and at least in part upon the customer preferences maintained in the customer preferences database for the particular customer;

communicate the calculated customer-preferred ship date to an order scheduler;

obtain a targeted ship date for the customer order entry from the order scheduler; and

calculate a customer expected delivery date by adding data representing anticipated shipping delay to the targeted ship date, taking into consideration data from the customer preferences database indicating whether the customer defines delivery dates to be ship dates or dock dates, and by also adding the customer's desired advance delivery time to the targeted ship date.

2. A computer system as recited in claim **1** and further programmed to specify preferred performance measurement species in the delivery and reporting preferences for individual customers, and to generate on-time product delivery statistics for individual customers in accordance with the individual customers' specified preferred performance measurement species.

3. A computer system as recited in claim **1** and further programmed to specify days-in-week variables in the delivery and reporting preferences for individual customers, and determine whether the product delivery is on-time based at least in part upon the customer's days-in-week variable.

4. A computer system as recited in claim **1** and further programmed to specify partial shipment allowed flags in the delivery and reporting preferences for individual customers, and count a partial delivery which is on time as a fraction of an on-time delivery if a customer's partial shipment allowed flag indicates that said customer allows partial on-time shipments to be counted in on-time statistics.

5. A computer system as recited in claim **1** and further programmed to specify reporting periods in the delivery and reporting preferences for individual customers, and generate on-time product delivery statistics periodically for a particular customer based upon that customer's specified reporting periods.

6. A computer system as recited in claim 1 and further programmed to obtain actual dock dates for individual product deliveries, wherein determining whether the particular product delivery is on time is based upon that delivery's actual dock date.

7. A computer system as recited in claim 1 and further programmed to obtain actual dock dates for individual product deliveries through electronic data interchange with carriers, wherein determining whether the particular product delivery is on time is based upon that delivery's actual dock date.

8. A method of setting and reporting product delivery dates comprising:

maintaining a customer preferences database having delivery and reporting preferences for individual customers, said preferences including preferred early and late delivery limits, preferred performance measurement species, desired advance delivery times, and data indicating whether the customer defines delivery dates to be ship dates or dock dates;

creating a customer order entry for a particular customer, said customer order entry including a customer-requested delivery date supplied by said particular customer;

calculating a customer-preferred ship date for the customer order entry based at least in part upon the customer-requested delivery date and at least in part upon the particular customer's specified desired advance delivery time;

routing the customer order entry to an order scheduler;

showing the order scheduler the calculated customer-preferred ship date;

obtaining a targeted ship date for the customer order entry from the order scheduler;

calculating a targeted ship date window based upon the targeted ship date and the particular customer's preferred early and late delivery limits; and

calculating a customer expected delivery date by adding data representing anticipated shipping delay to the targeted ship date, taking into consideration data from the customer preferences database indicating whether the customer defines delivery dates to be ship dates or dock dates, and by also adding the customer's desired advance delivery time to the targeted ship date.

9. A method of setting and reporting product delivery dates as recited in claim 8 and further comprising generating on-time product delivery statistics for individual customers in accordance with the individual customers' specified preferred performance measurement species, said step of generating the on-time product delivery statistics including a further step of determining whether a particular product delivery is on time based upon the individual customers' preferred early and late delivery limits.

10. A method of setting and reporting product delivery dates as recited in claim 8 and further comprising specifying days-in-week variables in the delivery and reporting preferences for individual customers, and determining whether the product delivery is on-time based at least in part upon the customer's days-in-week variable.

11. A method of setting and reporting product delivery dates as recited in claim 8 and further comprising specifying partial shipment allowed flags in the delivery and reporting preferences for individual customers, and counting a partial delivery which is on time as a fraction of an on-time delivery if a customer's partial shipment allowed flag indicates that said customer allows partial on-time shipments to be counted in on-time statistics.

12. A method of setting and reporting product delivery dates as recited in claim 8 and further comprising specifying reporting periods in the delivery and reporting preferences for individual customers, and generating on-time product delivery statistics periodically for a particular customer based upon that customer's specified reporting periods.

13. A method of setting and reporting product delivery dates as recited in claim 8 and further comprising obtaining actual dock dates for individual product deliveries.

14. An on-time delivery tracking and reporting system for use with a customer preferences database having delivery and reporting preferences for individual customers, said preferences including preferred early and late delivery limits, desired advance delivery times, and data for individual customers to indicate whether the customers consider delivery dates for particular products to be ship dates or dock dates, the system including a memory bearing software to:

create a customer order entry for a particular customer, said customer order entry including a customer-requested delivery date supplied by said particular customer;

calculate a customer-preferred ship date for the customer order entry based at least in part upon the customer-requested delivery date and at least in part upon the customer preferences maintained in the customer preferences database for the particular customer;

show the calculated customer-preferred ship date to an order scheduler;

obtain a targeted ship date for the customer order entry from the order scheduler;

calculate a customer expected delivery date by adding data representing anticipated shipping delay to the targeted ship date, taking into consideration data from the customer preferences database indicating whether the customer defines delivery dates to be ship dates or dock dates, and by also adding the customer's desired advance delivery time to the targeted ship date; and

determine whether a particular product delivery is on time, based upon the particular customer's preferred early and late delivery limits.

15. An on-time delivery tracking and reporting system as recited in claim 14 wherein the customer preferences database includes days-in-week variables in the delivery and reporting preferences for individual customers, and the software is configured to determine whether the product delivery is on-time based at least in part upon the customer's days-in-week variable.

16. An on-time delivery tracking and reporting system as recited in claim 14 wherein the customer preferences database includes partial shipment allowed flags in the delivery and reporting preferences for individual customers, and the software is configured to count a partial delivery which is on time as a fraction of an on-time delivery if a customer's partial shipment allowed flag indicates that said customer allows partial on-time shipments to be counted in on-time statistics.

17. An on-time delivery tracking and reporting system as recited in claim 14 wherein the customer preferences database includes reporting periods in the delivery and reporting preferences for individual customers, and the software is configured to generate on-time product delivery statistics periodically for a particular customer based upon that customer's specified reporting periods.

18. An on-time delivery tracking and reporting system as recited in claim 14 wherein the customer preferences database includes desired advance delivery times in the delivery

and reporting preferences for individual customers, and the software is configured to calculate the customer-preferred ship date based at least in part upon the particular customer's specified desired advance delivery time.

19. An on-time delivery tracking and reporting system as recited in claim **14** wherein the customer preferences database includes actual dock dates for individual product deliveries, and the software is configured to determine whether the particular product delivery is on time based upon that delivery's actual dock date.

20. An on-time delivery tracking and reporting system comprising:

a customer preferences database having delivery and reporting preferences for individual customers, said preferences including preferred early and late delivery limits, preferred performance measurement species, and desired advance delivery times; and

a data processor connected to access the customer preferences database and being programmed to:

create a customer order entry for a particular customer, said customer order entry including a customer-requested delivery date supplied by said particular customer; and

calculate a customer-preferred ship date for the customer order entry based at least in part upon the customer-requested delivery date and at least in part upon the particular customer's specified desired advance delivery time from the customer preferences database.

21. An on-time delivery tracking and reporting system including a memory bearing software to:

determine and store in a database of customer preferences data representing a customer's early and late delivery limits for deliveries;

determine and store in the customer preferences database whether the customer defines delivery date as shipped date or received date for deliveries;

determine and store in the customer preferences database whether the customer includes weekends and holidays in evaluating whether a delivery is on-time;

determine and store how performance is measured by the customer, whether by dollars, by units, by line items, by shipments;

determine and store in the customer preferences database the customer's desired advance delivery time;

determine and store in the customer preferences database whether the customer considers a partial shipment in determining on-time delivery;

determine and store the customer's preferred reporting period;

periodically prompt a user to update the preferences;

receive an order from the customer for a particular delivery, and receive a requested delivery date;

access the customer preferences database and calculate the customer's preferred ship date as being the customer's requested delivery date minus the customer's desired advance delivery;

send the customer's preferred ship date to a human order scheduler;

receive a targeted ship date from the human order scheduler;

calculate a customer expected delivery date by adding data representing anticipated shipping delay to the targeted ship date, taking into consideration data from the customer preferences database indicating whether the customer defines delivery dates to be ship dates or dock dates, and by also adding the customer's desired advance delivery time to the targeted ship date;

receive the actual delivery date for the particular delivery;

generate delivery statistics according to the customer's reporting period, taking into consideration data from the customer preferences database including how performance is measured by the customer, whether by dollars, by units, by line items, by shipments, including whether the customer defines delivery dates to be ship dates or dock dates, and including whether the customer includes weekends and holidays in evaluating whether a delivery is on-time; and

record reasons for deliveries which were not on time.

22. A method of setting and reporting product delivery dates comprising:

maintaining a customer preferences database having delivery and reporting preferences for individual customers, said preferences including preferred early and late delivery limits, preferred performance measurement species, and desired advance delivery times;

specifying reporting periods in the delivery and reporting preferences for individual customers;

creating a customer order entry for a particular customer, said customer order entry including a customer-requested delivery date supplied by said particular customer;

calculating a customer-preferred ship date for the customer order entry based at least in part upon the customer-requested delivery date and at least in part upon the particular customer's specified desired advance delivery time;

routing the customer order entry to an order scheduler;

showing the order scheduler the calculated customer-preferred ship date;

obtaining a targeted ship date for the customer order entry from the order scheduler;

specifying ship/dock flags in the delivery and reporting preferences for individual customers to indicate whether said customers consider delivery dates for particular products to be ship dates or dock dates;

calculating a customer expected delivery date by adding data representing anticipated shipping delay to the targeted ship date, taking into consideration whether the particular customer considers a delivery date to be a ship date or dock date, and by also adding the customer's desired advance delivery time to the targeted ship date;

specifying days-in-week variables in the delivery and reporting preferences for individual customers;

specifying partial shipment allowed flags in the delivery and reporting preferences for individual customers;

obtaining actual dock dates for individual product deliveries; and

generating on-time product delivery statistics for individual customers in accordance with the individual customers' specified preferred performance measurement species and based upon that customer's specified reporting periods.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO : 5,960,408
DATED : September 28, 1999
INVENTOR(S) : Doreen J. Martin et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Col. 3, line 25
replace "colendar"
with --calendar--.

Signed and Sealed this

Twenty-fifth Day of April, 2000

Attest:

Todd

Q. TODD DICKINSON

Attesting Officer

Director of Patents and Trademarks

US005832459A

United States Patent [19]

Cameron et al.

[11] **Patent Number: 5,832,459**

[45] **Date of Patent: Nov. 3, 1998**

[54] **COMPUTERIZED SOURCE SEARCHING SYSTEM AND METHOD FOR USE IN AN ORDER ENTRY SYSTEM**

[75] Inventors: **Paul Scott Cameron**, Minneapolis; **John Charles Nash**, Shoreview; **Robert Christopher Bloomer**, Little Canada; **Robert Edward Wollan**, Minneapolis; **Kelly Marie Kreutter**, Minnetonka; **Melinda Ann Ahler Olmstead**, Shoreview; **Dale Harry Renner**, Edina; **Ryan Douglas Bourne**, Eden Prairie; **Keith Michael Carnish**, Minneapolis; **Dean Richard Jones**, St. Louis Park, all of Minn.

[73] Assignee: **Andersen Consulting LLP**, Chicago, Ill.

[21] Appl. No.: **696,779**

[22] Filed: **Aug. 13, 1996**

Related U.S. Application Data

[62] Division of Ser. No. 293,470, Aug. 19, 1994, Pat. No. 5,592,378.

[51] **Int. Cl.**^{6} **G06F 17/60**

[52] **U.S. Cl.** **705/26**; 705/27

[58] **Field of Search** 705/1, 14, 26, 705/27; 707/1, 2, 3, 4, 5, 6, 10, 100, 104

[56] **References Cited**

U.S. PATENT DOCUMENTS

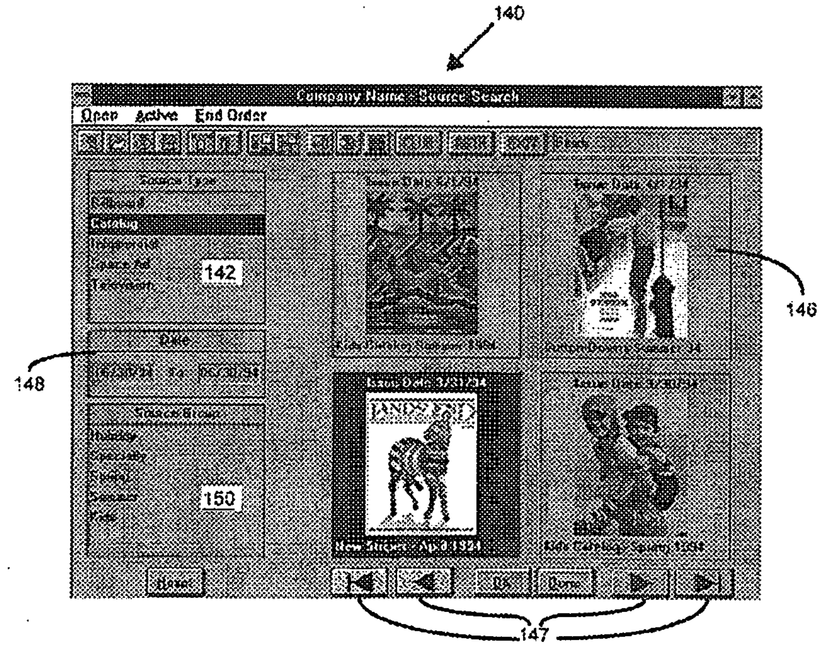
4,490,810	12/1984	Hon	463/43
4,734,858	3/1988	Schlafly	235/380
4,775,935	10/1988	Yourick	345/357
4,803,348	2/1989	Lohrey et al.	235/381
4,887,208	12/1989	Schneider	705/28
4,984,155	1/1991	Geier et al.	705/26
4,992,940	2/1991	Dworkin	705/26
5,146,404	9/1992	Calloway	705/1
5,168,445	12/1992	Kawashima et al.	705/10
5,241,464	8/1993	Greulich et al.	705/26
5,283,731	2/1994	Lalonde et al.	705/1
5,309,355	5/1994	Lockwood	705/6
5,319,542	6/1994	King, Jr. et al.	705/27

Primary Examiner—Stephen R. Tkacs
Attorney, Agent, or Firm—Merchant, Gould, Smith, Edell, Welter & Schmidt, P.A.

[57] **ABSTRACT**

A computerized source searching system and method for the placement of an order for at least one offer via a terminal having a display is disclosed. The system includes system storage for storing offer information, and for storing electronic reproductions of offer sources corresponding to the offer information. The system includes source searching capabilities for locating an offer source, including a display for displaying electronic offer source images, a filtering mechanism for displaying electronic offer source images corresponding to entered source search criteria, and a selection mechanism for selecting one of the electronic offer source images displayed. One or more of the offers associated with the selected electronic offer source image can be located by displaying a segment of the electronic offer source image, and selecting one or more offers in that segment. A user can thereby locate an offer by searching through an electronic equivalent of the offer source used by the person placing the order.

34 Claims, 41 Drawing Sheets



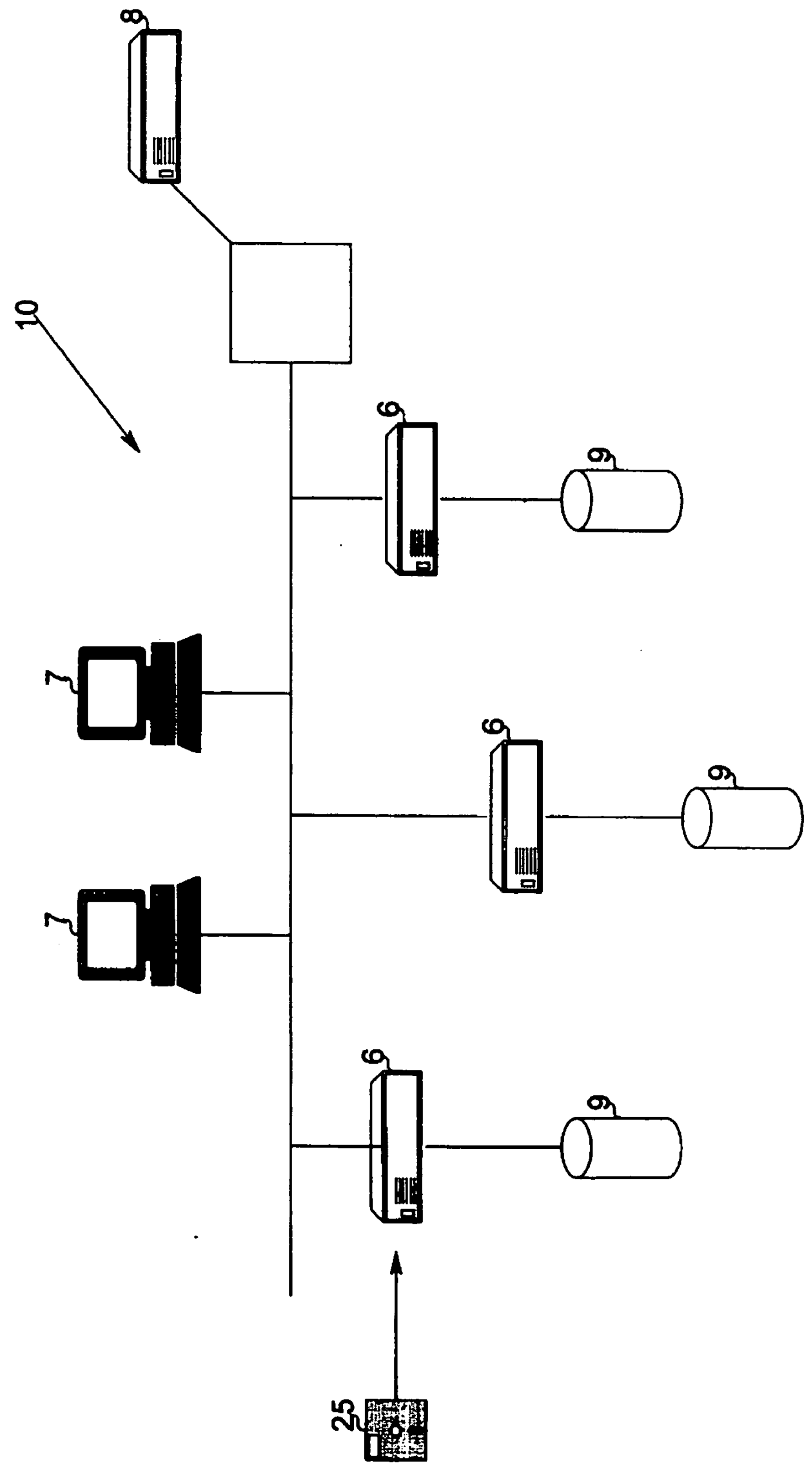
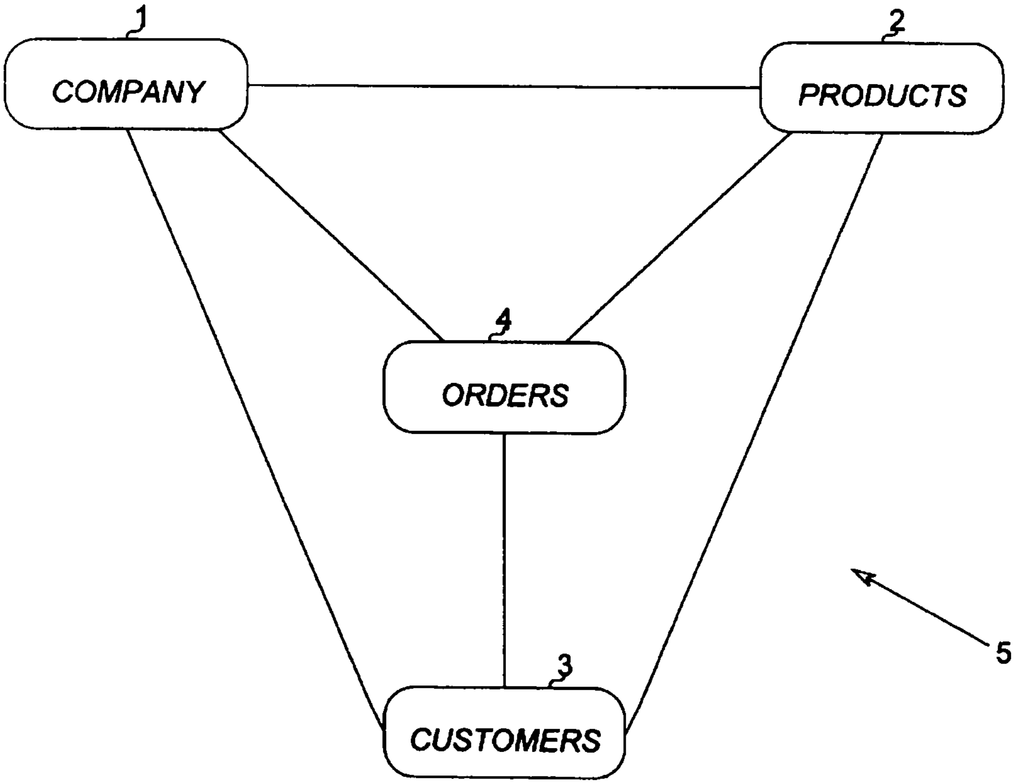


FIG. 1

FIG. 2



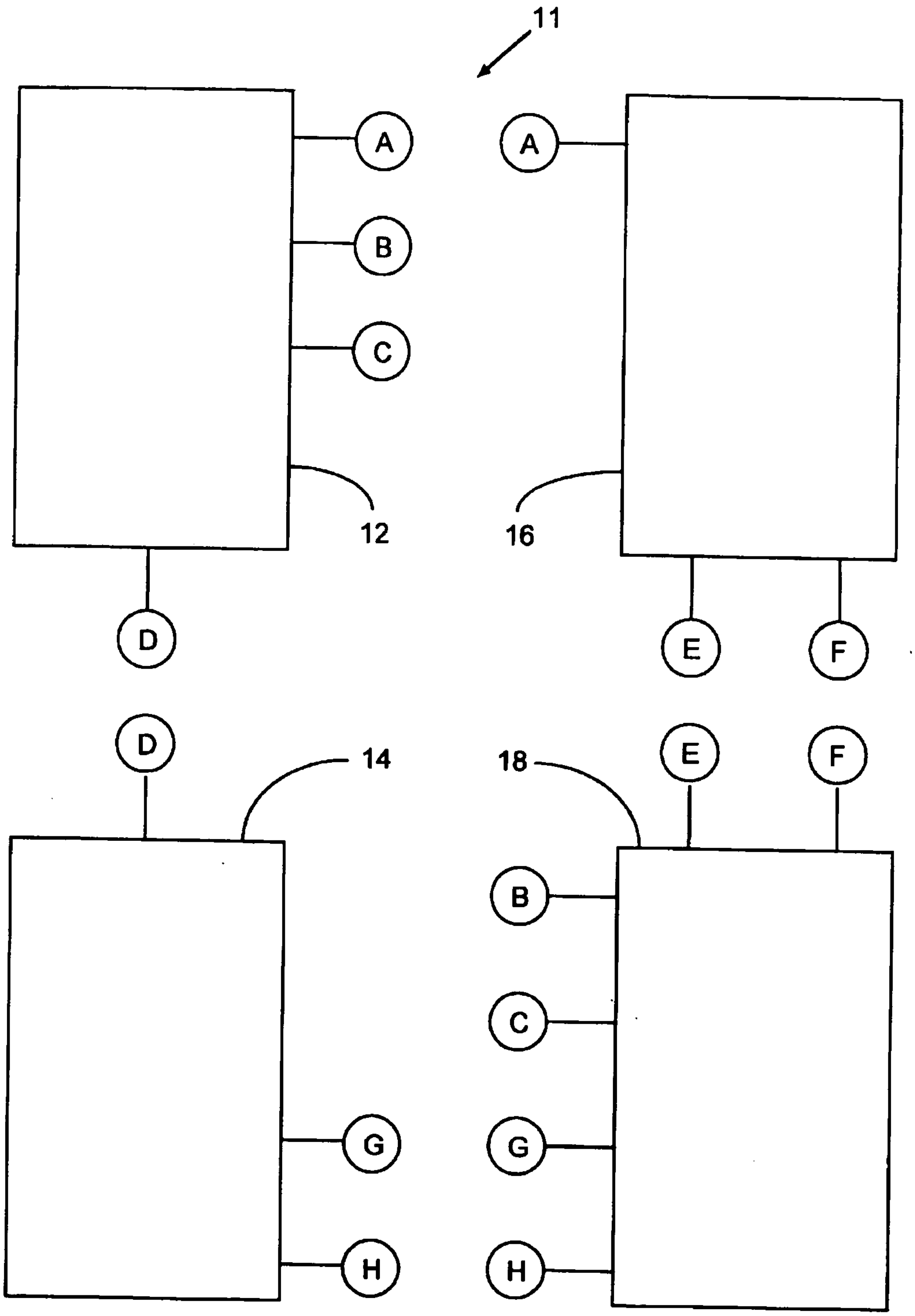
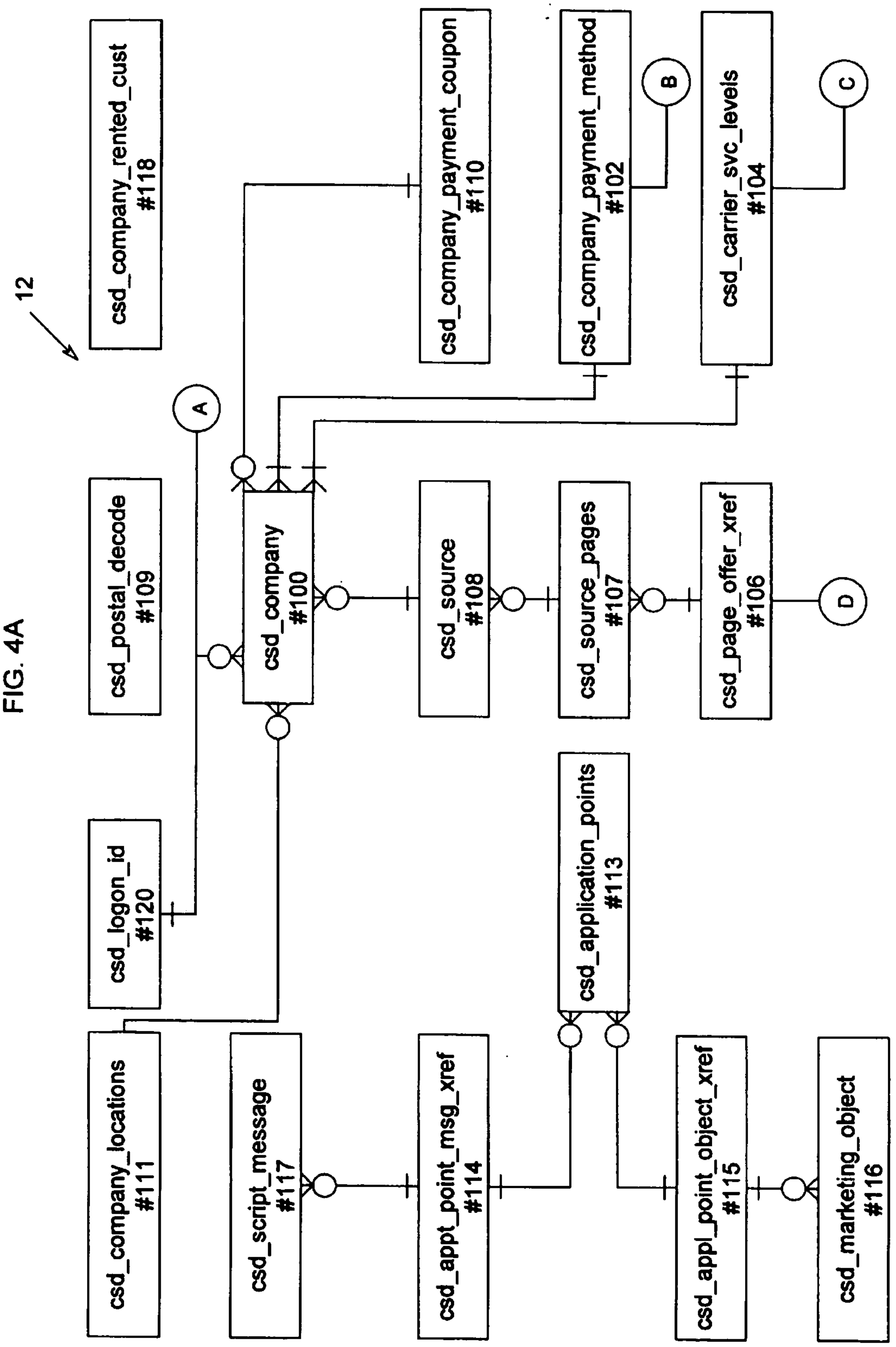


FIG. 3

FIG. 4A


12
csd_company_rented_cust
#118
csd_company_payment_coupon
#110
csd_company_payment_method
#102
B
csd_carrier_svc_levels
#104
C
A
csd_postal_decode
#109
csd_company
#100
csd_source
#108
csd_source_pages
#107
csd_page_offer_xref
#106
D
csd_logon_id
#120
csd_application_points
#113
csd_company_locations
#111
csd_script_message
#117
csd_appt_point_msg_xref
#114
csd_appl_point_object_xref
#115
csd_marketing_object
#116

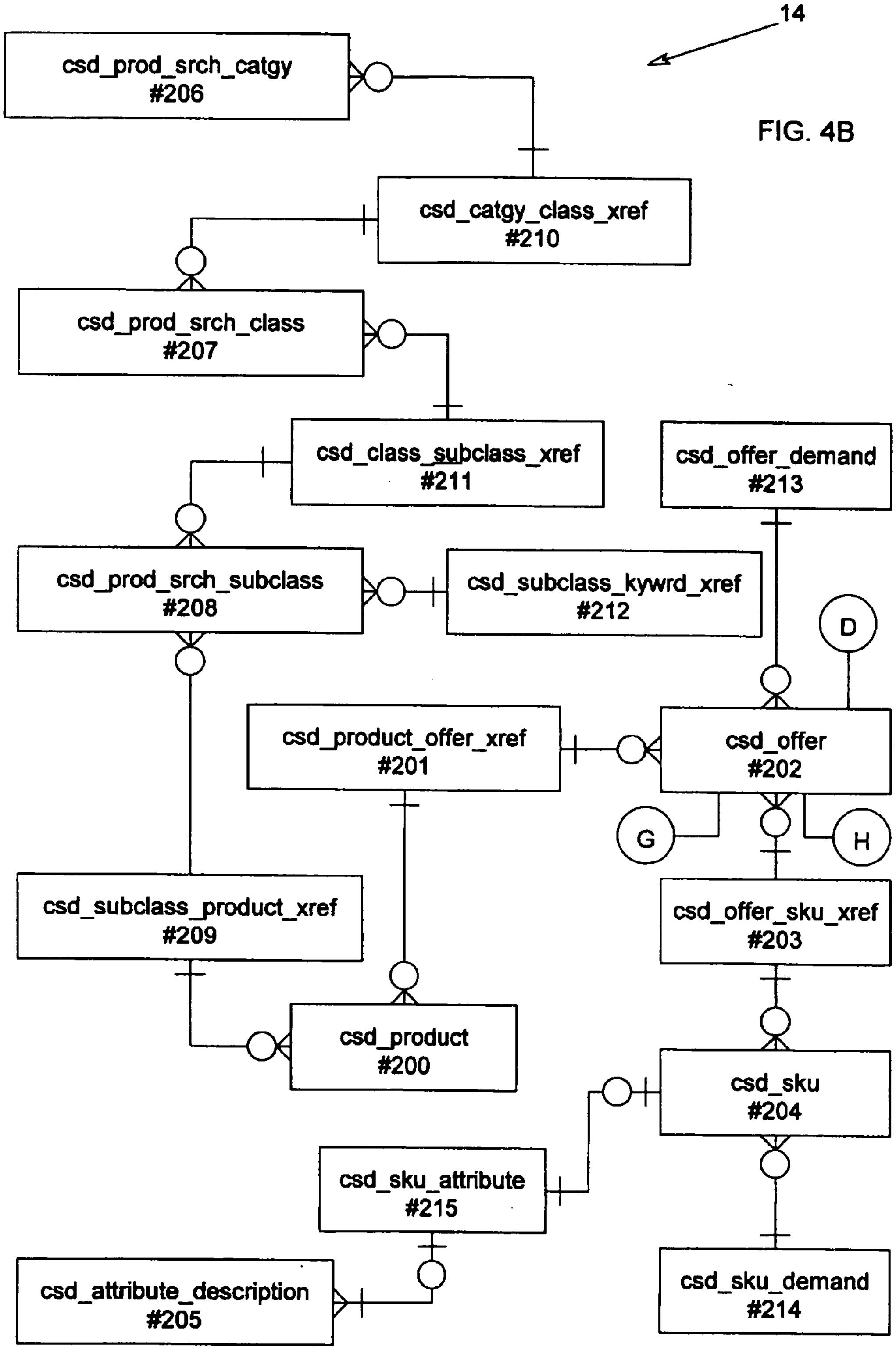
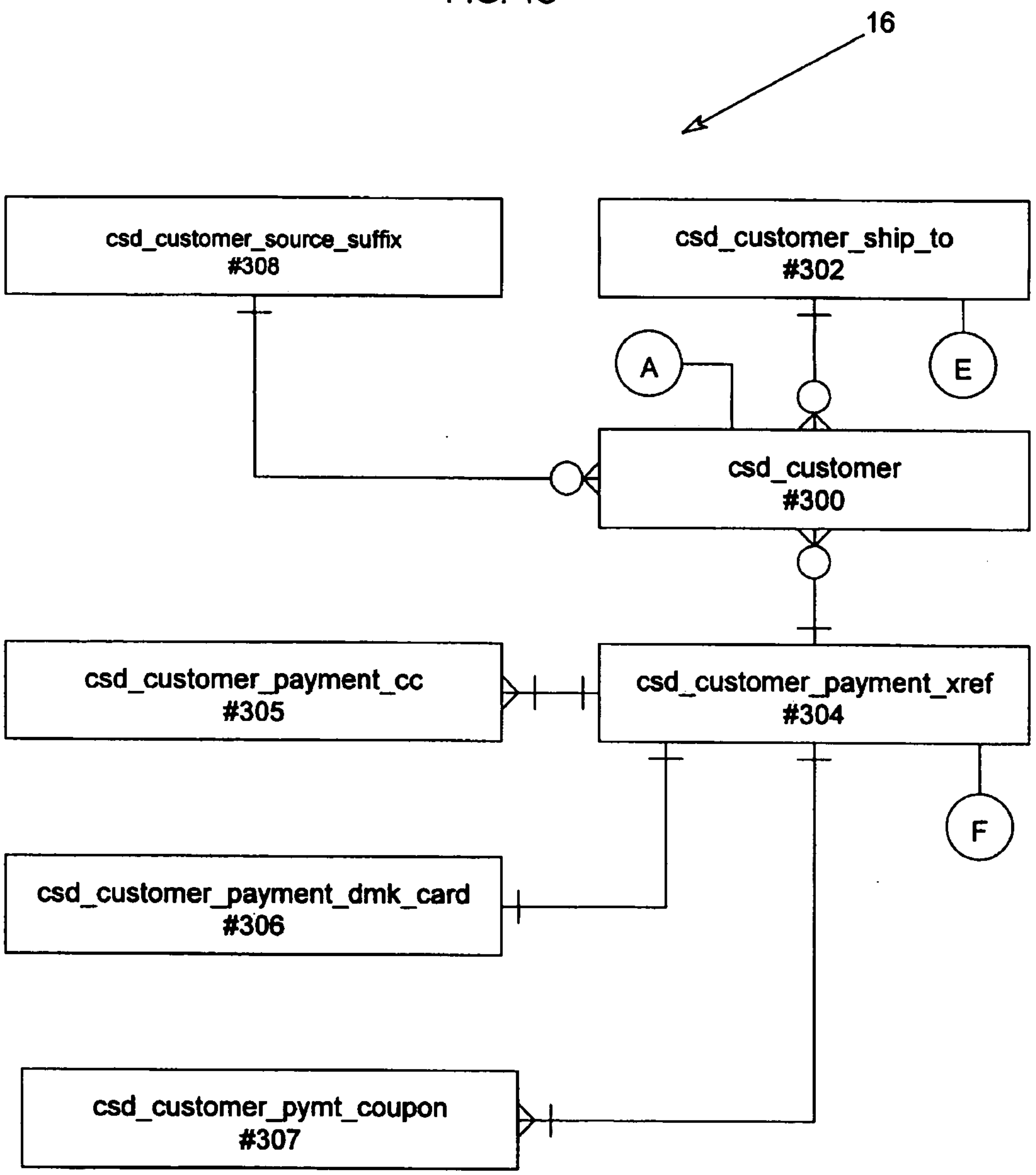

14
csd_prod_srch_catgy
#206
csd_catgy_class_xref
#210
csd_prod_srch_class
#207
csd_class_subclass_xref
#211
csd_offer_demand
#213
csd_prod_srch_subclass
#208
csd_subclass_kywrd_xref
#212
D
csd_product_offer_xref
#201
csd_offer
#202
G
H
csd_subclass_product_xref
#209
csd_offer_sku_xref
#203
csd_product
#200
csd_sku
#204
csd_sku_attribute
#215
csd_attribute_description
#205
csd_sku_demand
#214

FIG. 4B

FIG. 4C


16
csd_customer_source_suffix
#308
csd_customer_ship_to
#302
A
E
csd_customer
#300
csd_customer_payment_cc
#305
csd_customer_payment_xref
#304
F
csd_customer_payment_dmk_card
#306
csd_customer_pymt_coupon
#307

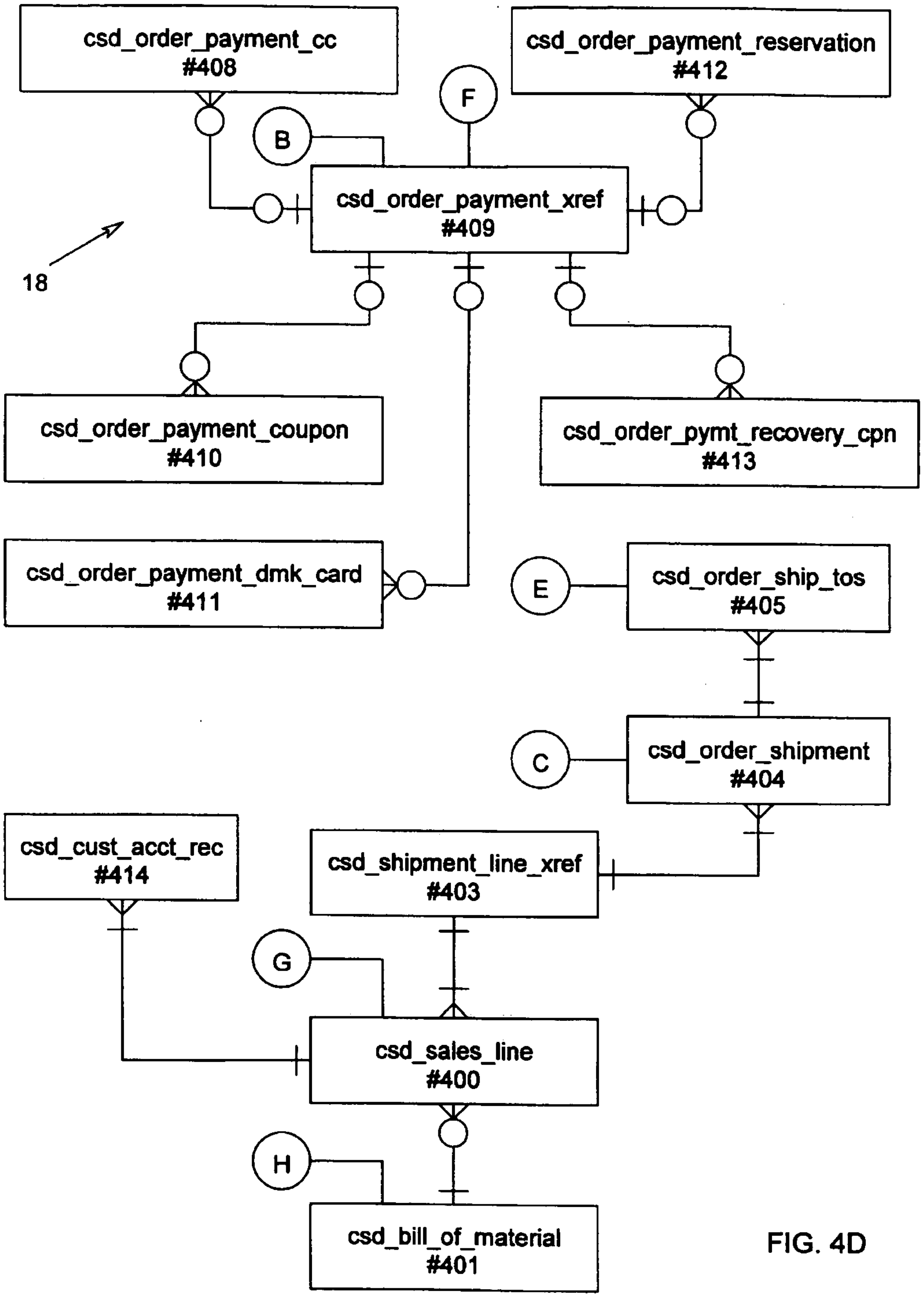

csd_order_payment_cc
#408
csd_order_payment_reservation
#412
F
B
csd_order_payment_xref
#409
18
csd_order_payment_coupon
#410
csd_order_pymt_recovery_cpn
#413
csd_order_payment_dmk_card
#411
E
csd_order_ship_tos
#405
C
csd_order_shipment
#404
csd_cust_acct_rec
#414
csd_shipment_line_xref
#403
G
csd_sales_line
#400
H
csd_bill_of_material
#401

FIG. 4D

FIG. 5A

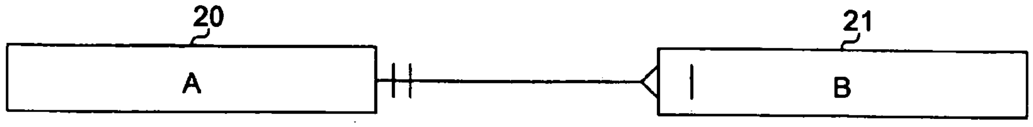


FIG. 5B

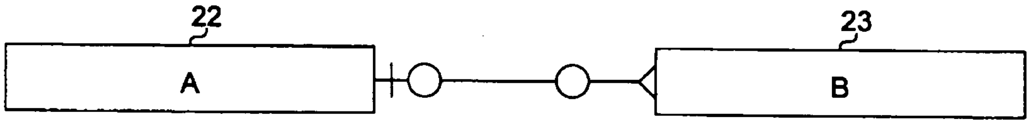
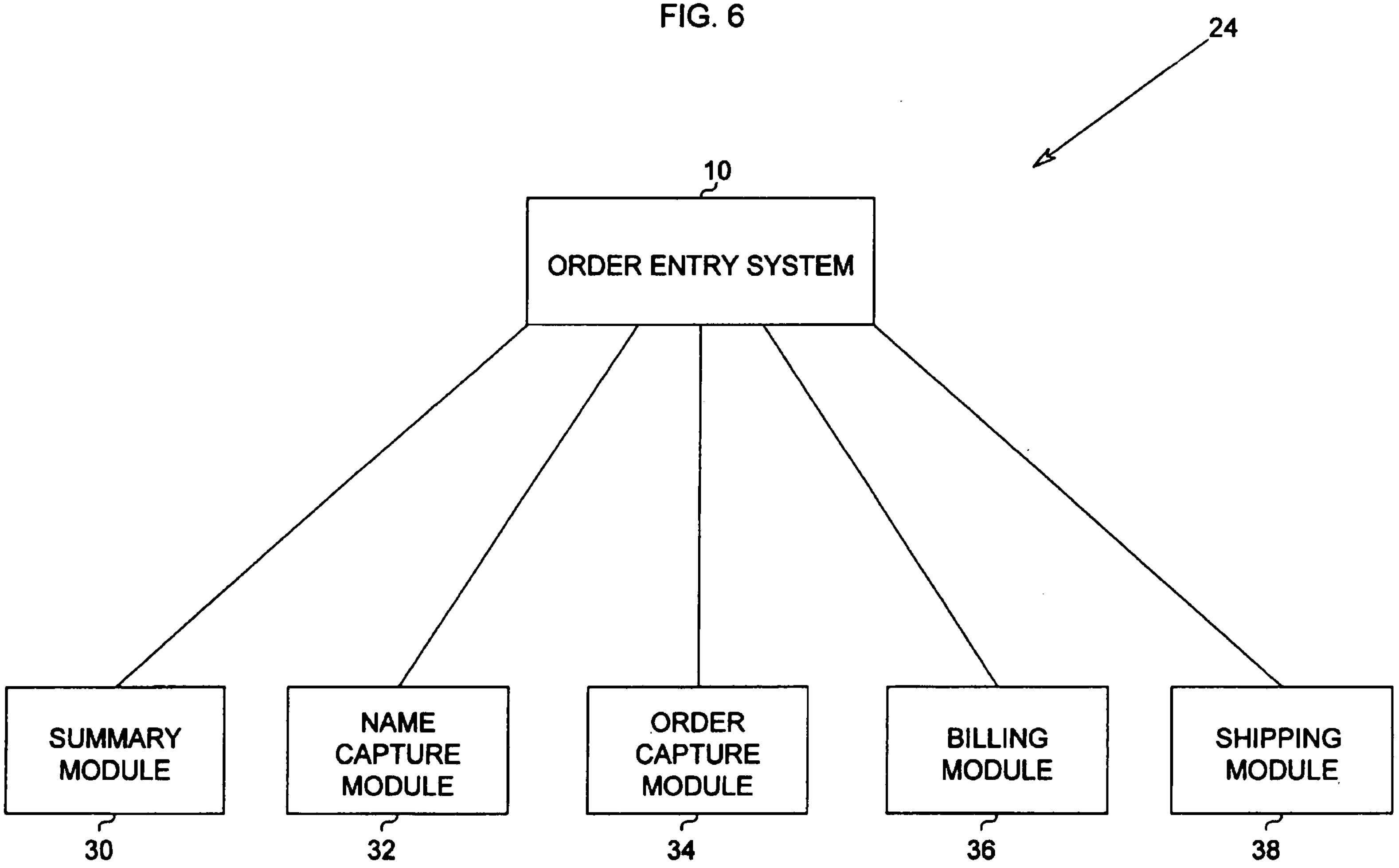
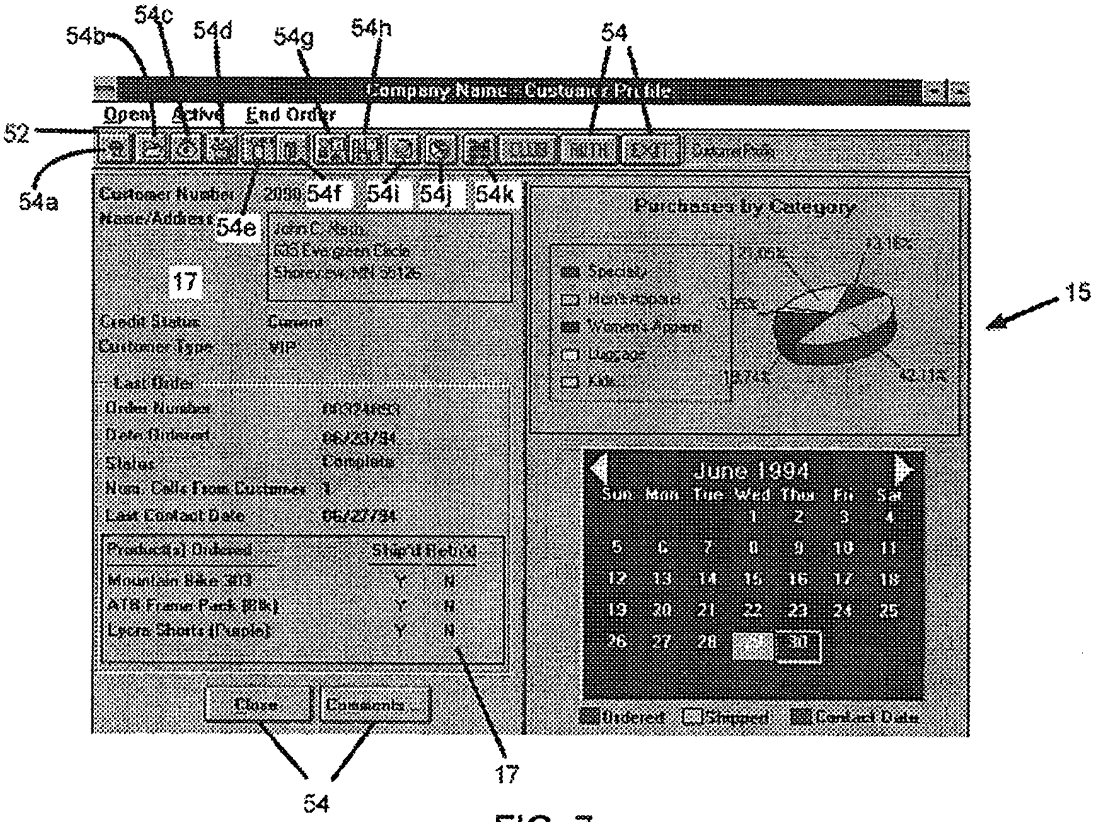


FIG. 6


24
10
ORDER ENTRY SYSTEM
SUMMARY MODULE
30
NAME CAPTURE MODULE
32
ORDER CAPTURE MODULE
34
BILLING MODULE
36
SHIPPING MODULE
38



Customer Number 2030

Name/Address John C. [illegible]
535 Evergreen Circle
Shoreview, MN 55126

Credit Status Current

Customer Type VIP

Last Order

Order Number 00094850

Date Ordered 06/23/94

Status Complete

Num. Calls From Customer 1

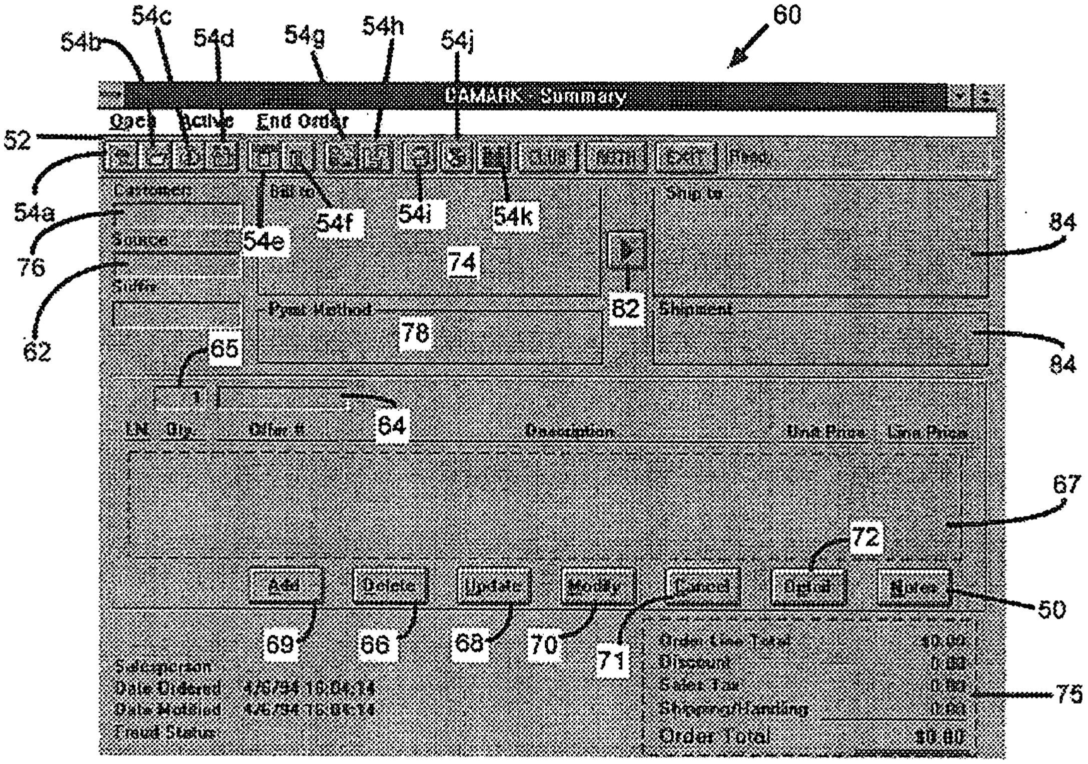
Last Contact Date 06/27/94

Product(s) Ordered	Ship'd	Return
Mountain Bike 303	Y	N
ATB Frame Pack (Blk)	Y	N
Lycra Shorts (Purple)	Y	N

June 1994

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

FIG. 7



Salesperson
Date Ordered 4/6/94 16:04:14
Date Modified 4/6/94 16:04:14
Fraud Status

Order Line Total	$0.00
Discount	0.00
Sales Tax	0.00
Shipping/Handling	0.00
Order Total	$0.00

FIG. 8

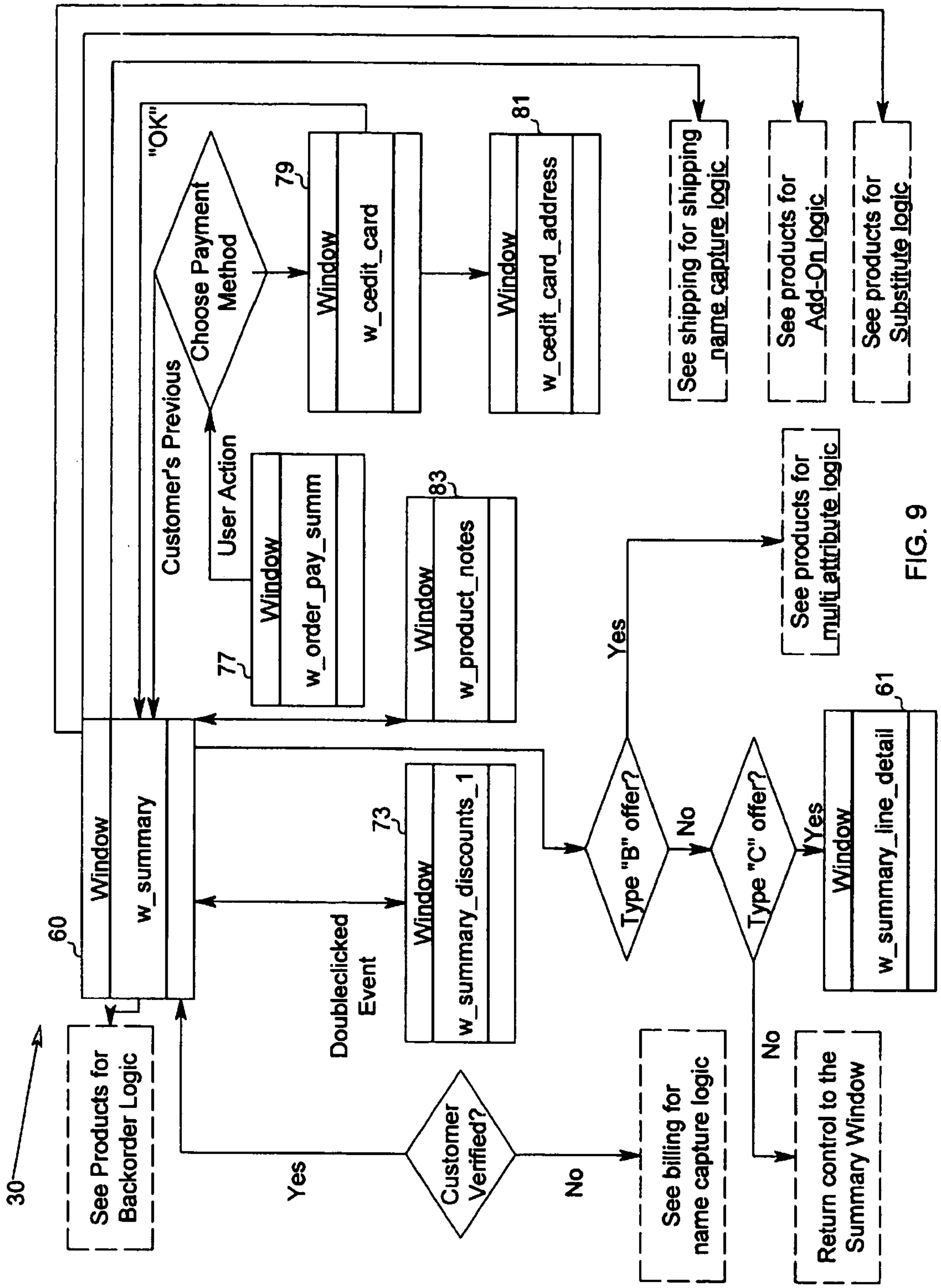
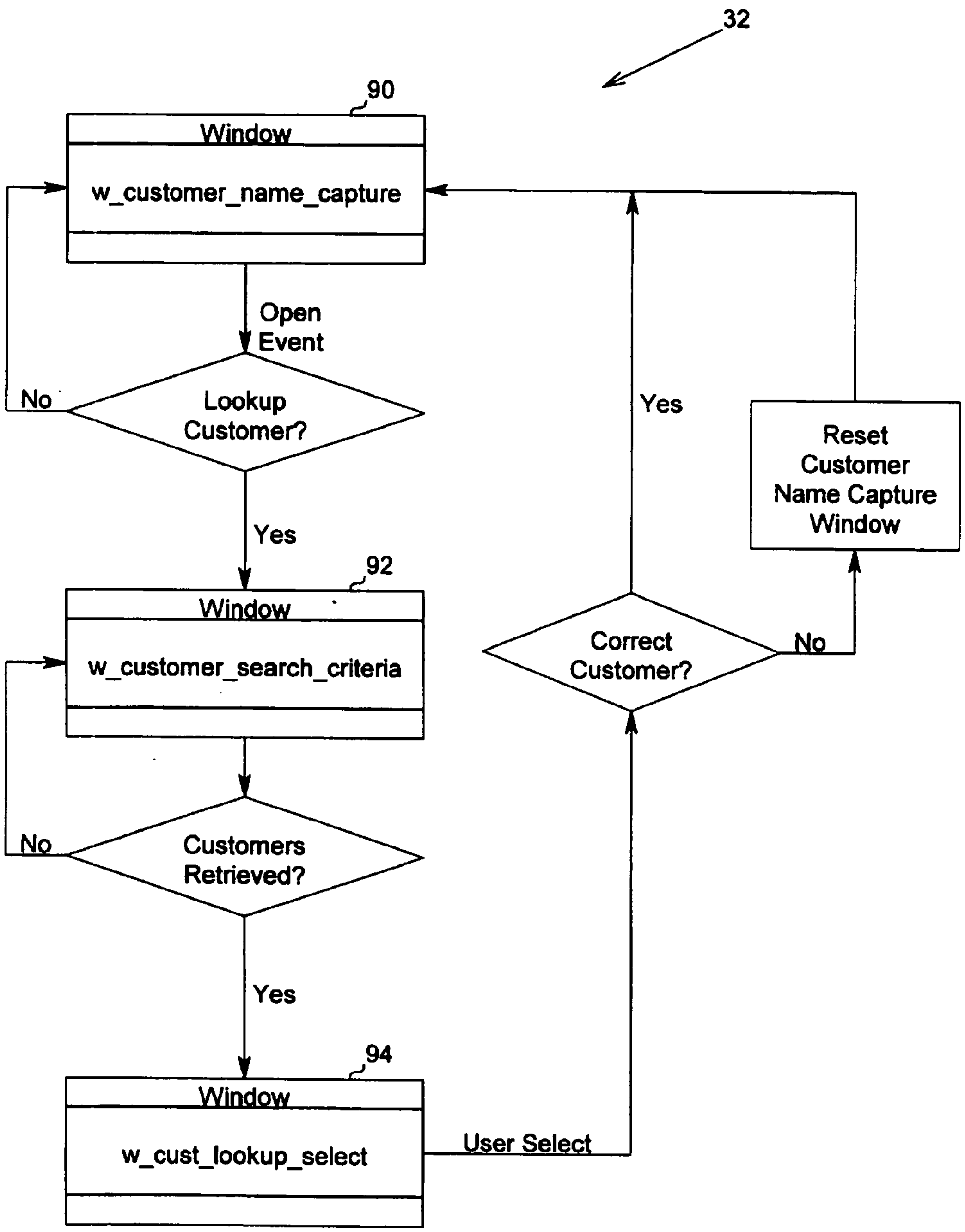
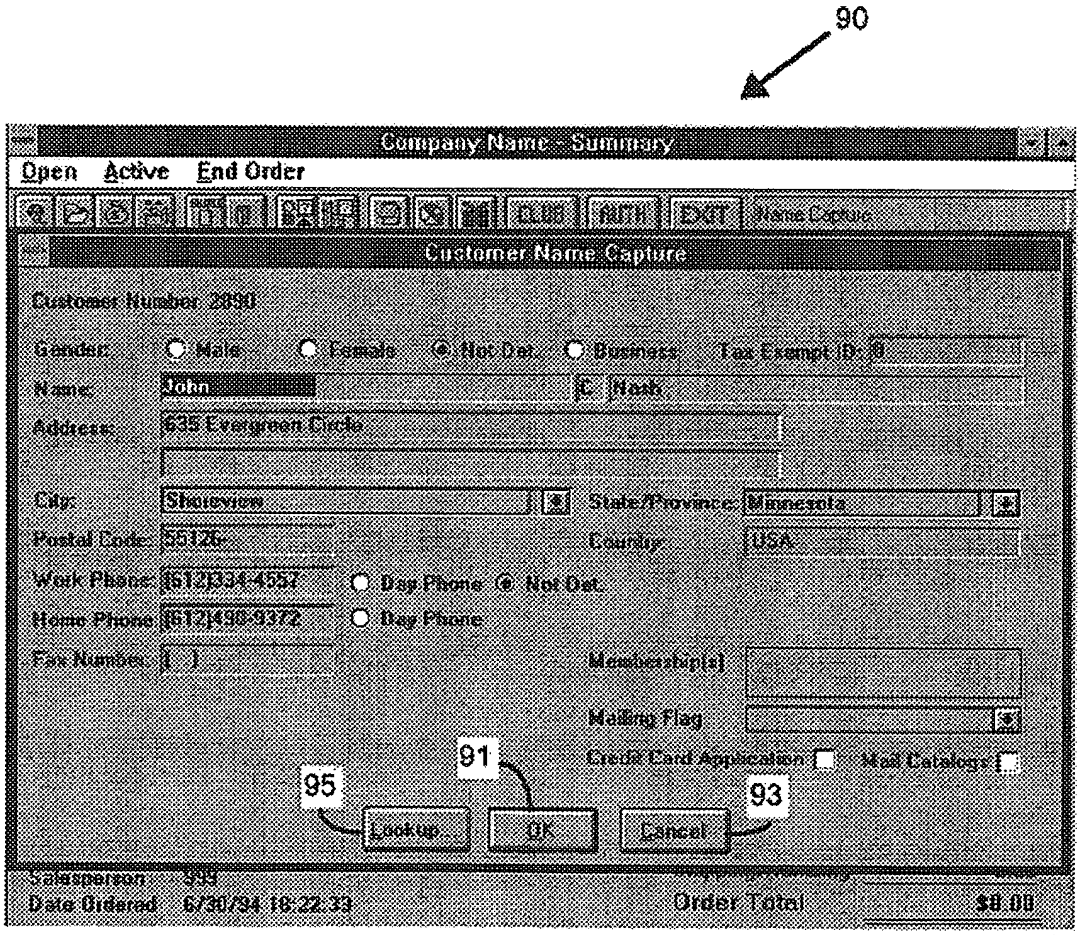

30
60
Window
w_summary
See Products for Backorder Logic
Customer's Previous
"OK"
Choose Payment Method
User Action
77
Window
w_order_pay_summ
79
Window
w_cedit_card
81
Window
w_cedit_card_address
83
Window
w_product_notes
See shipping for shipping name capture logic
See products for Add-On logic
See products for Substitute logic
Doubleclicked Event
73
Window
w_summary_discounts_1
Type "B" offer?
Yes
See products for multi attribute logic
No
Type "C" offer?
Yes
61
Window
w_summary_line_detail
No
Return control to the Summary Window
Customer Verified?
Yes
No
See billing for name capture logic

FIG. 9

FIG. 10


32
90
Window
w_customer_name_capture
Open
Event
No
Lookup
Customer?
Yes
92
Window
w_customer_search_criteria
No
Customers
Retrieved?
Yes
94
Window
w_cust_lookup_select
User Select
Correct
Customer?
Yes
No
Reset
Customer
Name Capture
Window



Customer Number 2890

Gender: ○ Male ○ Female ● Not Def. ○ Business Tax Exempt ID: 0

Name: John C Nash

Address: 635 Evergreen Circle

City: Shoreview State/Province: Minnesota

Postal Code: 55126- Country: USA

Work Phone: (612)334-4557 ○ Day Phone ● Not Def.

Home Phone (612)490-9372 ○ Day Phone

Fax Number: ( )

Membership(s)

Mailing Flag

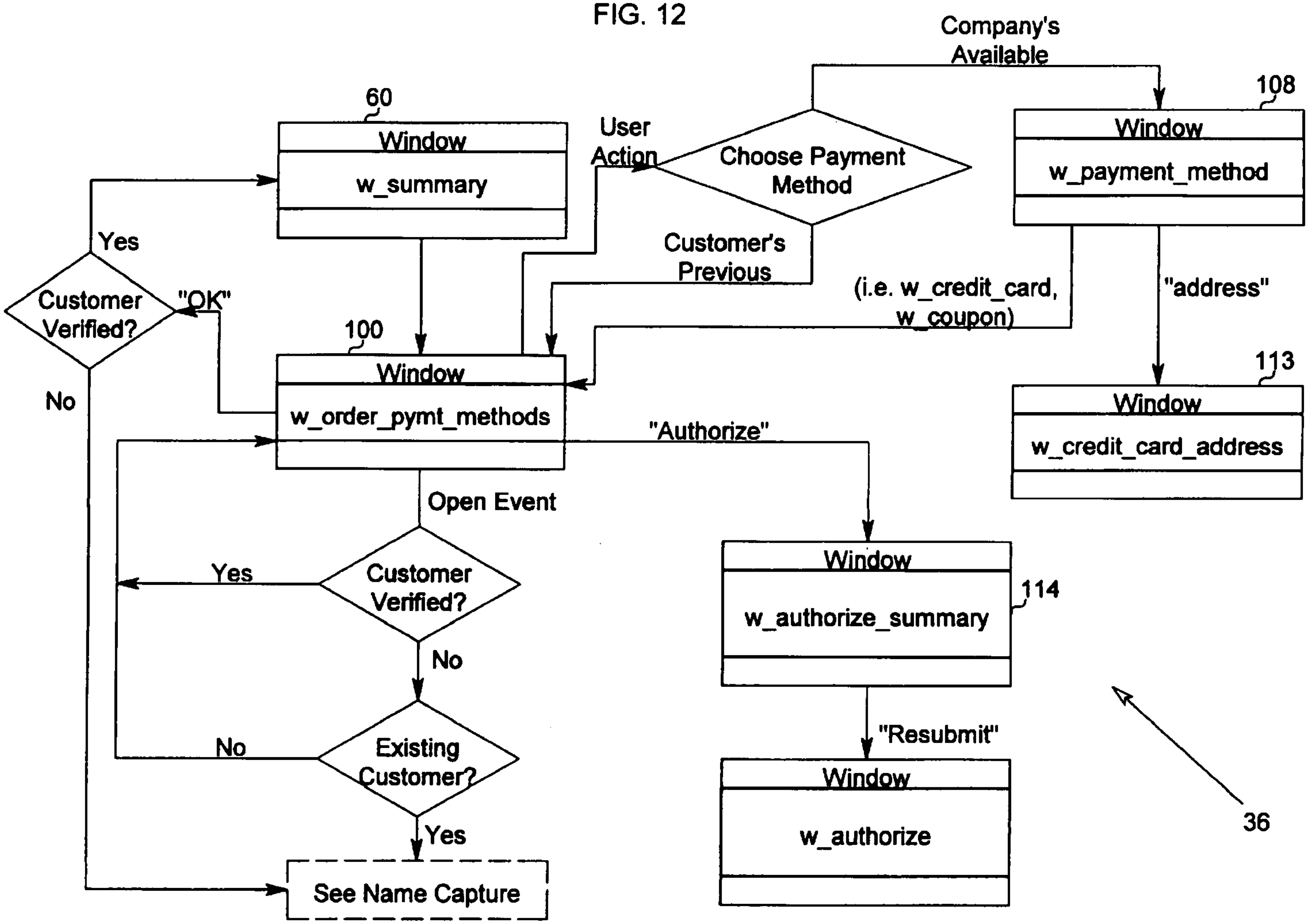
Credit Card Application □ Mail Catalogs □

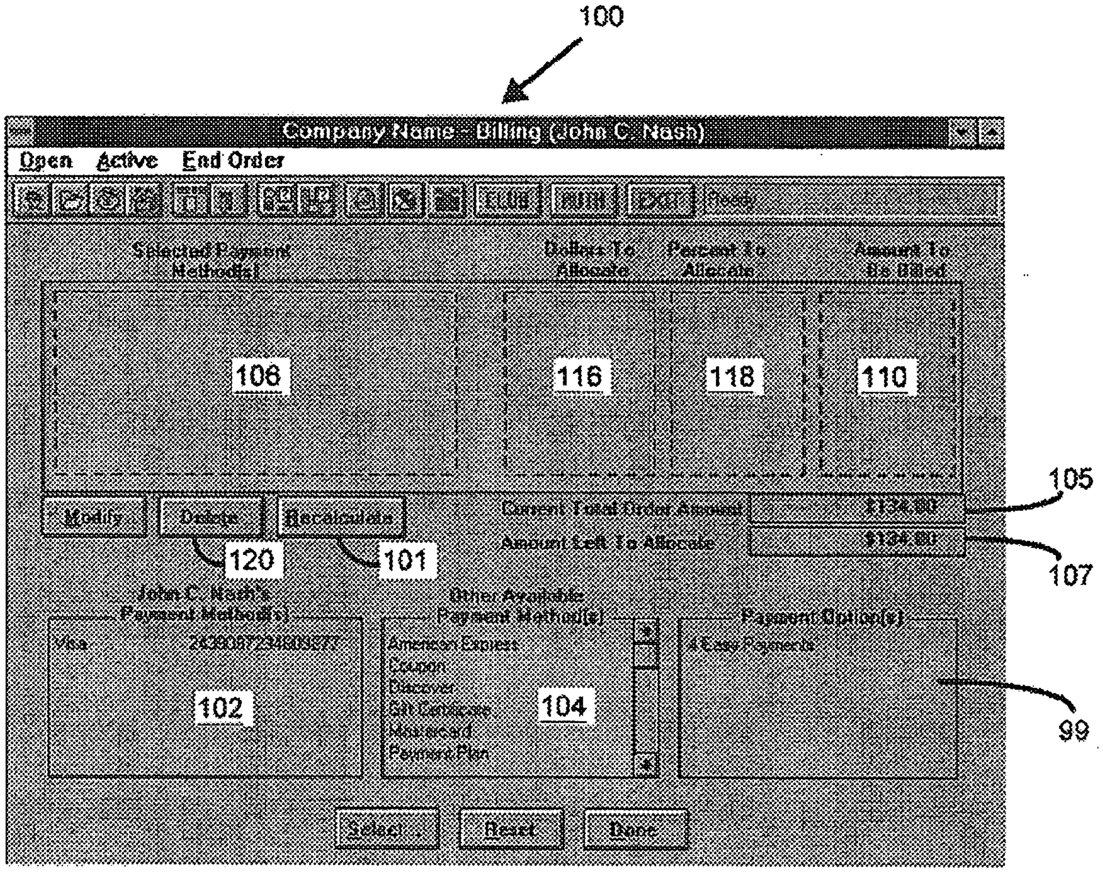
Salesperson 999

Date Ordered 6/30/94 18:22:33 Order Total $0.00

FIG. 11

FIG. 12


Company's Available
108
Window
w_payment_method
"address"
113
Window
w_credit_card_address
60
Window
w_summary
User Action
Choose Payment Method
Customer's Previous
(i.e. w_credit_card, w_coupon)
Yes
Customer Verified?
"OK"
No
100
Window
w_order_pymt_methods
"Authorize"
Open Event
Customer Verified?
Yes
No
Existing Customer?
No
Yes
See Name Capture
Window
w_authorize_summary
114
"Resubmit"
Window
w_authorize
36



Current Total Order Amount $134.80

Amount Left To Allocate $134.80

John C. Nash's Payment Method(s)

Visa 2439087234803277

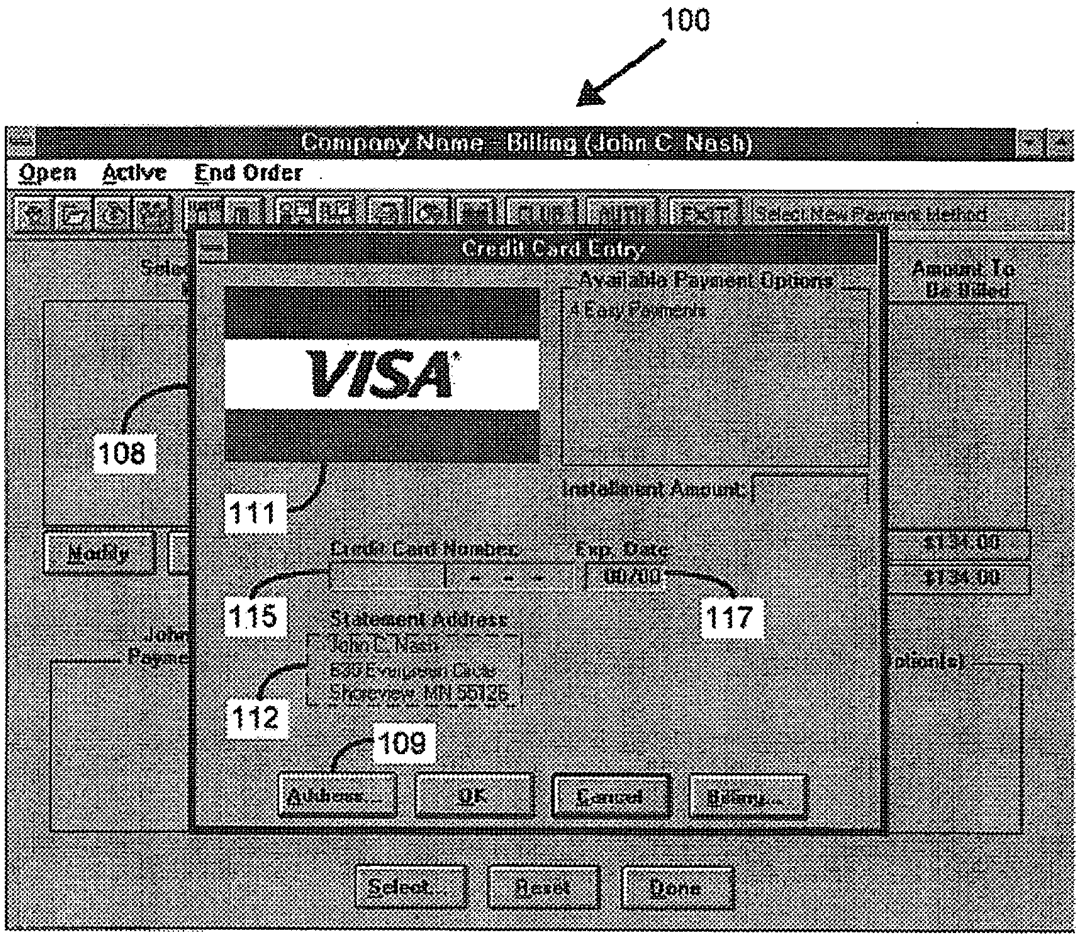
Other Available Payment Method(s)

American Express
Coupon
Discover
Gift Certificate
Mastercard
Payment Plan

Payment Option(s)

4 Easy Payments

FIG. 13



Company Name - Billing (John C. Nash)

Credit Card Entry

Available Payment Options
4 Easy Payments

Installment Amount:

Credit Card Number:

Exp. Date: 00/00

Statement Address
John C. Nash
830 Evergreen Circle
Shoreview, MN 55126

Amount To Be Billed
$134.00
$134.00

FIG. 14

FIG. 15

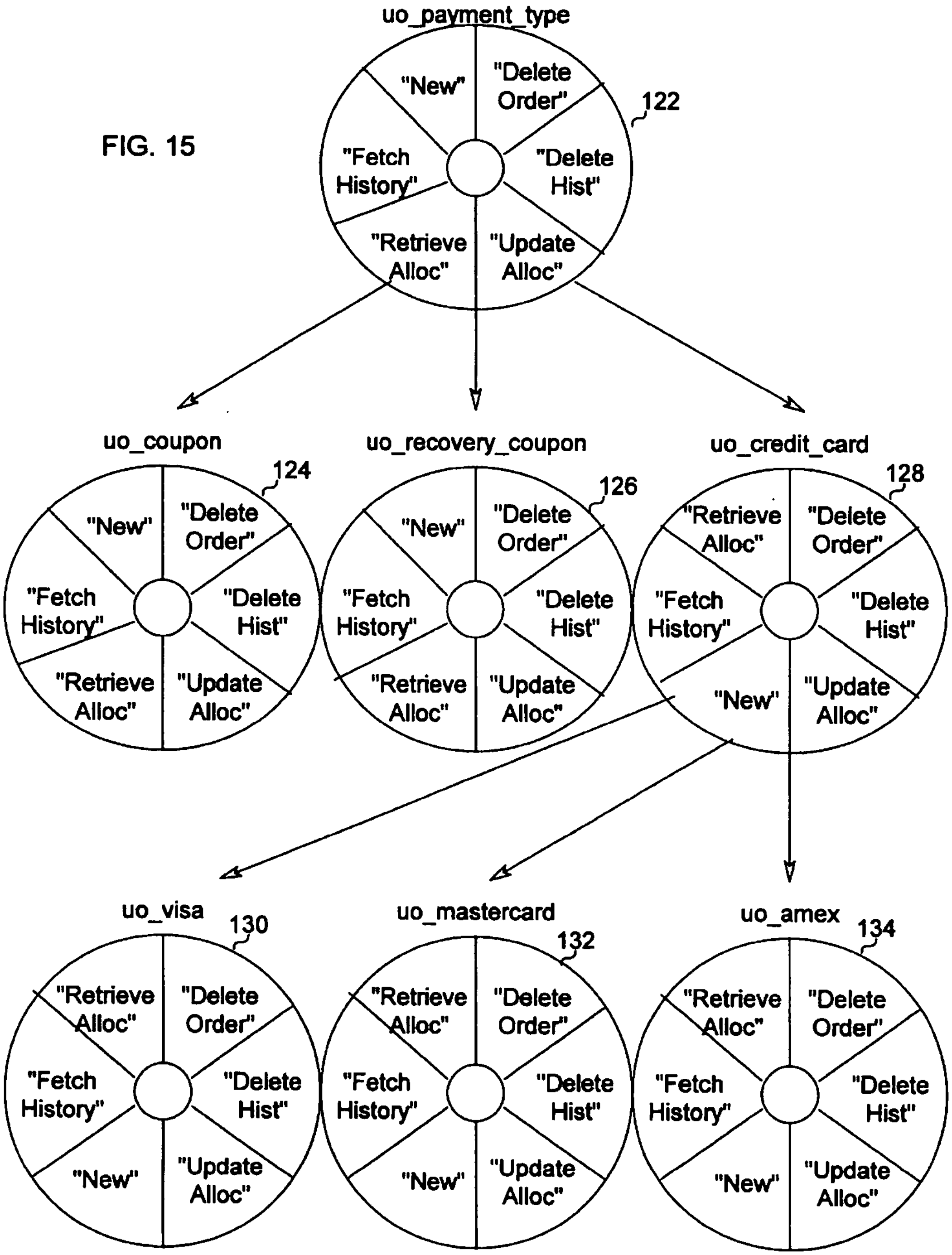

uo_payment_type
"New"
"Delete Order"
122
"Fetch History"
"Delete Hist"
"Retrieve Alloc"
"Update Alloc"
uo_coupon
124
"New"
"Delete Order"
"Fetch History"
"Delete Hist"
"Retrieve Alloc"
"Update Alloc"
uo_recovery_coupon
126
"New"
"Delete Order"
"Fetch History"
"Delete Hist"
"Retrieve Alloc"
"Update Alloc"
uo_credit_card
128
"Retrieve Alloc"
"Delete Order"
"Fetch History"
"Delete Hist"
"New"
"Update Alloc"
uo_visa
130
"Retrieve Alloc"
"Delete Order"
"Fetch History"
"Delete Hist"
"New"
"Update Alloc"
uo_mastercard
132
"Retrieve Alloc"
"Delete Order"
"Fetch History"
"Delete Hist"
"New"
"Update Alloc"
uo_amex
134
"Retrieve Alloc"
"Delete Order"
"Fetch History"
"Delete Hist"
"New"
"Update Alloc"

FIG. 16

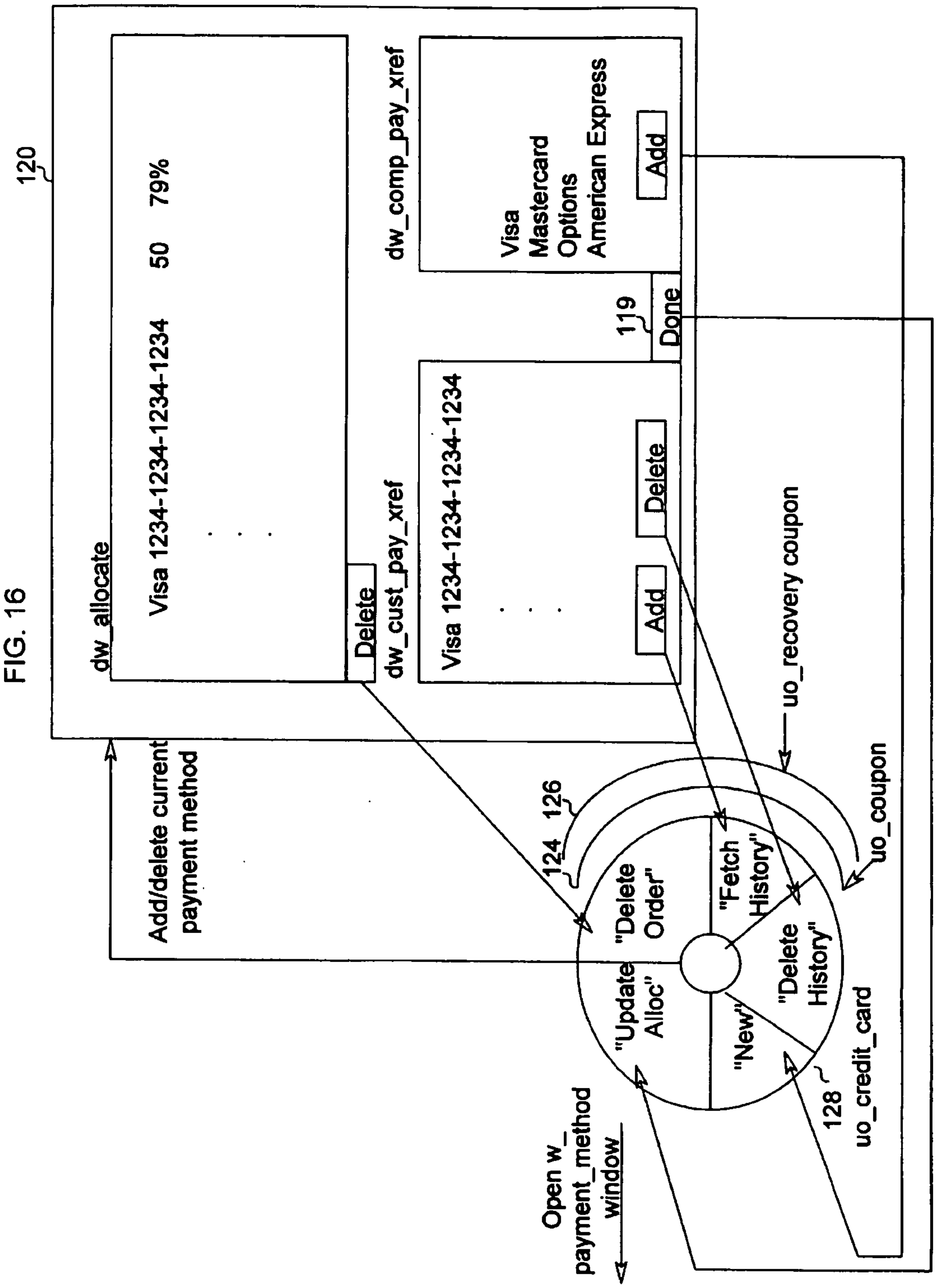

120
dw_allocate
Visa 1234-1234-1234-1234 50 79%
. . .
Delete
dw_cust_pay_xref
Visa 1234-1234-1234-1234
. . .
Add
Delete
119
Done
dw_comp_pay_xref
Visa
Mastercard
Options
American Express
Add
uo_recovery coupon
uo_coupon
124
126
"Delete Order"
"Fetch History"
"Delete History"
"New"
"Update Alloc"
128
uo_credit_card
Add/delete current payment method
Open w_payment_method window

FIG. 17A

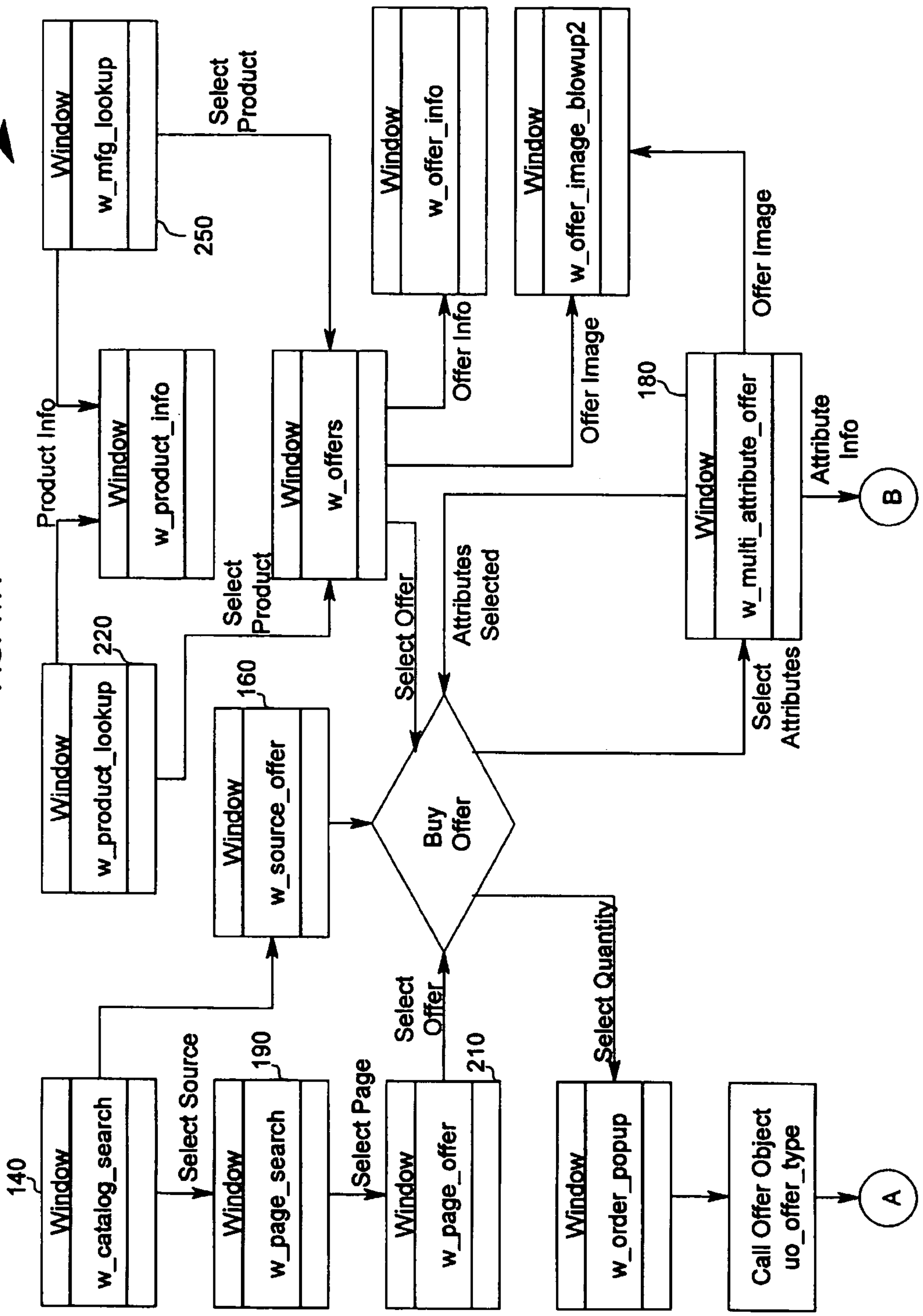
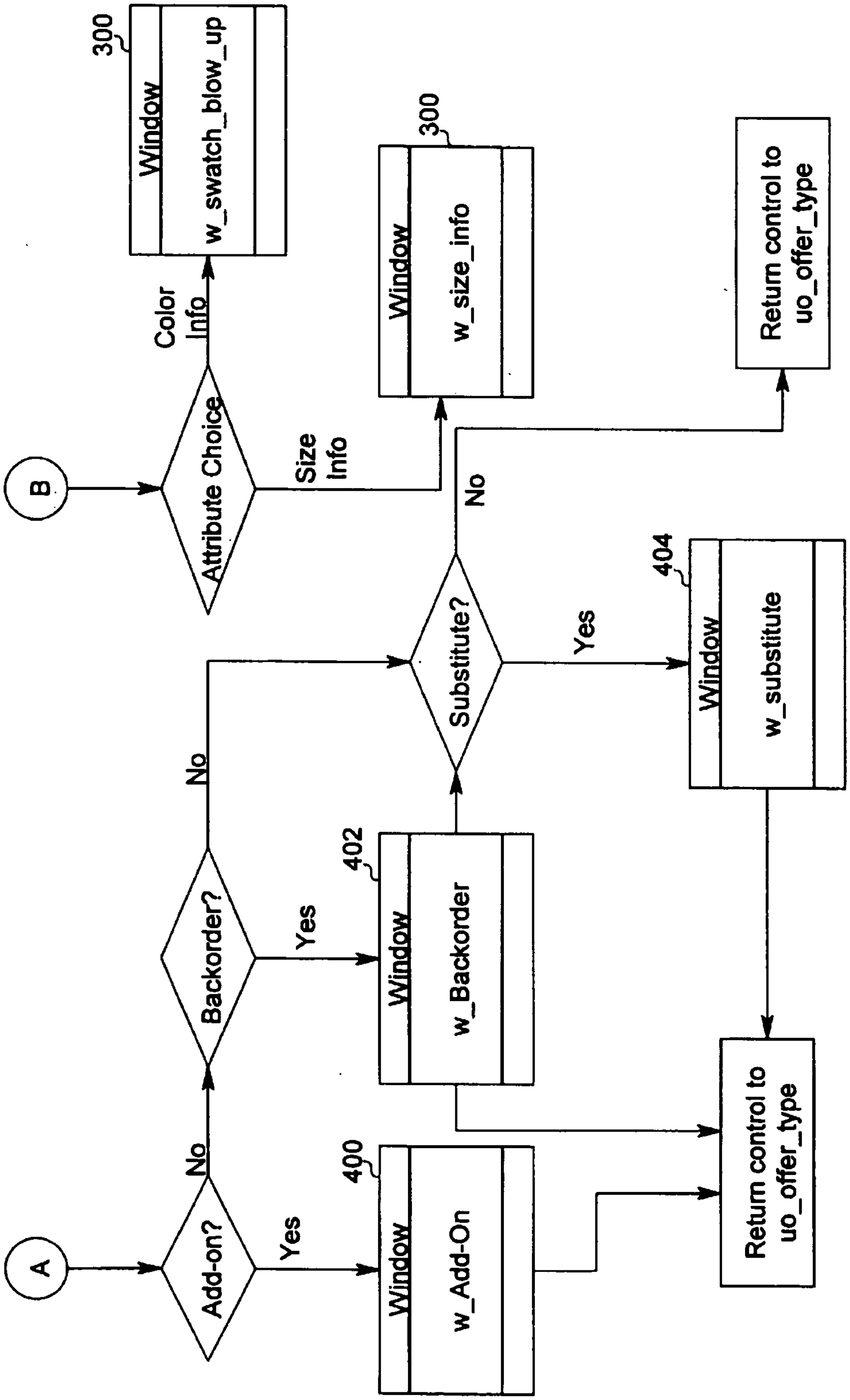
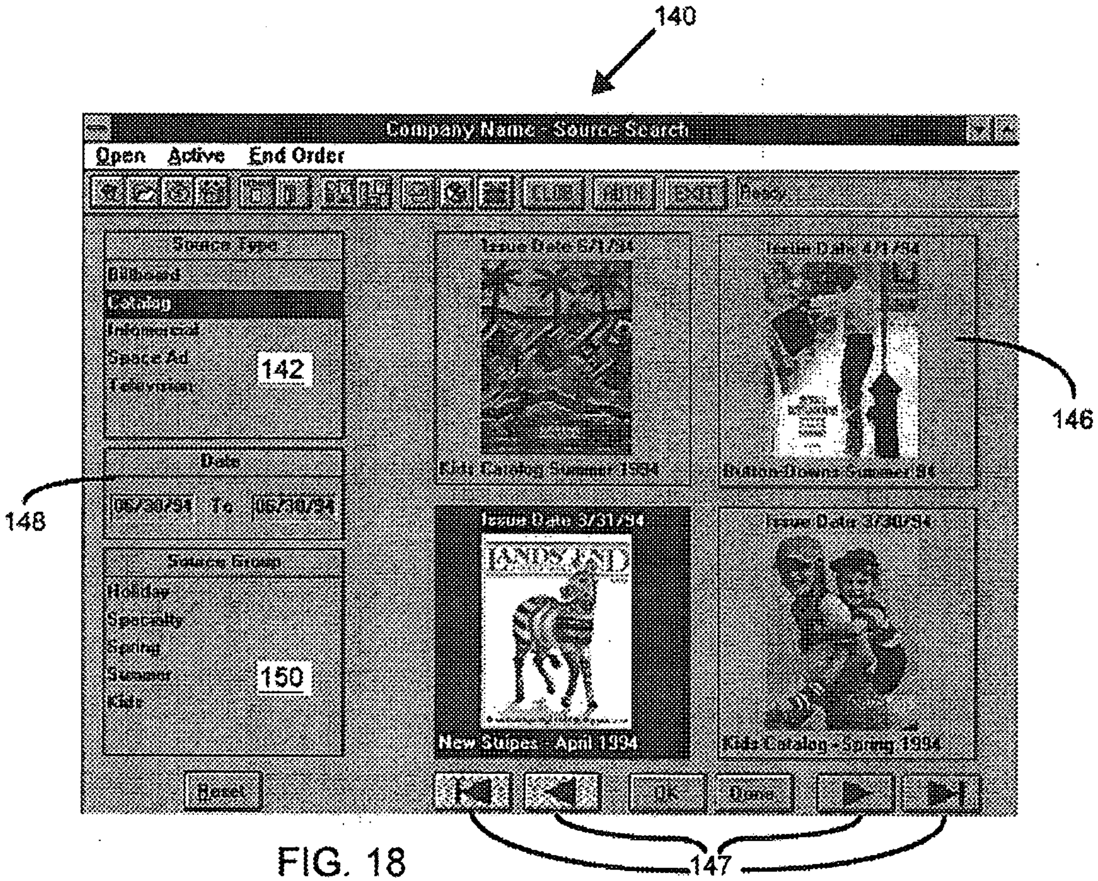

34
Window
w_mfg_lookup
250
Select Product
Window
w_offer_info
Window
w_offer_image_blowup2
Product Info
Window
w_product_info
Window
w_offers
Offer Info
Offer Image
180
Window
w_multi_attribute_offer
Offer Image
Attribute Info
B
220
Window
w_product_lookup
Select Product
Select Offer
Attributes Selected
160
Window
w_source_offer
Buy Offer
Select Attributes
140
Window
w_catalog_search
Select Source
190
Window
w_page_search
Select Page
Select Offer
210
Window
w_page_offer
Select Quantity
Window
w_order_popup
Call Offer Object
uo_offer_type
A

FIG. 17B


A
Add-on?
Yes
No
400
Window
w_Add-On
Backorder?
Yes
No
402
Window
w_Backorder
Substitute?
Yes
No
404
Window
w_substitute
Return control to uo_offer_type
Return control to uo_offer_type
B
Attribute Choice
Color Info
Size Info
300
Window
w_swatch_blow_up
300
Window
w_size_info



Source Type

Billboard
Catalog
Infomercial
Space Ad
Television

Date

06/30/94 To 06/30/94

Source Group

Holiday
Specialty
Spring
Summer
Kids

Issue Date 6/1/94
Kids Catalog Summer 1994

Issue Date 4/1/94
Button-Downs Summer 94

Issue Date 3/31/94
New Stripes - April 1994

Issue Date 3/30/94
Kids Catalog - Spring 1994

FIG. 18

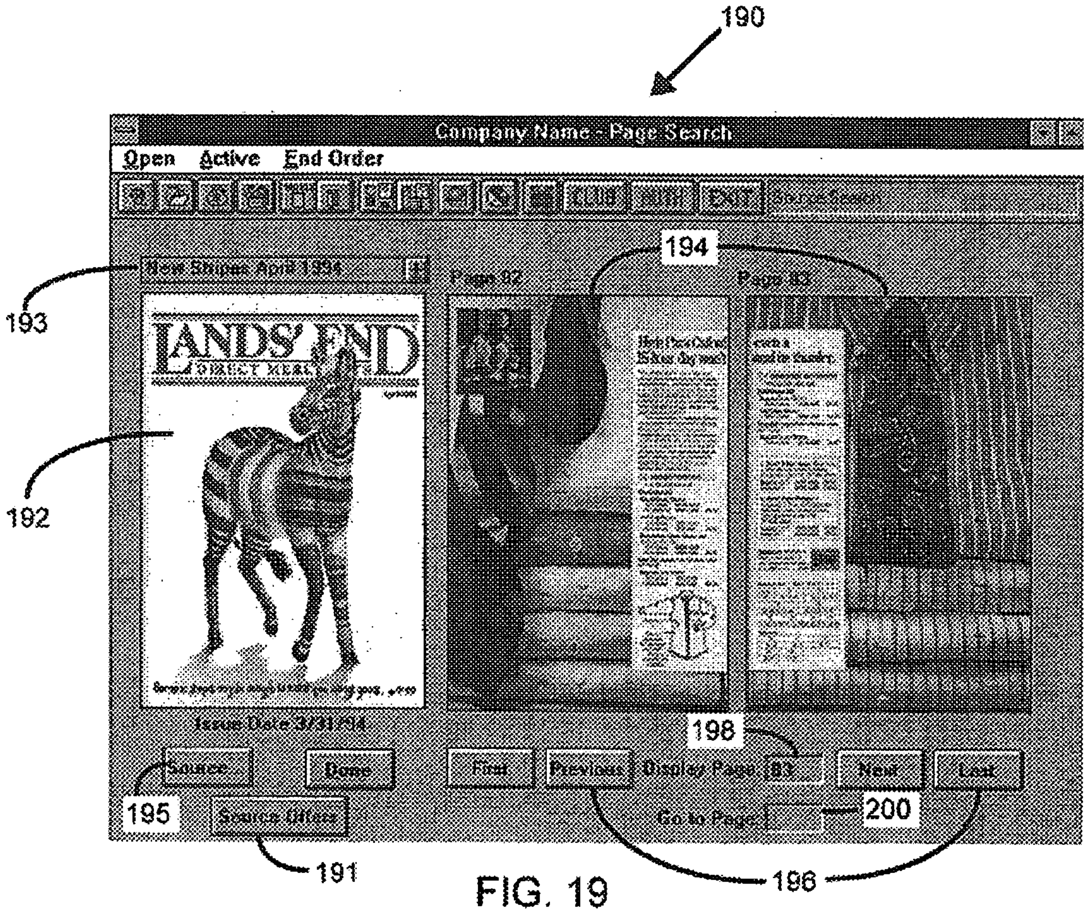
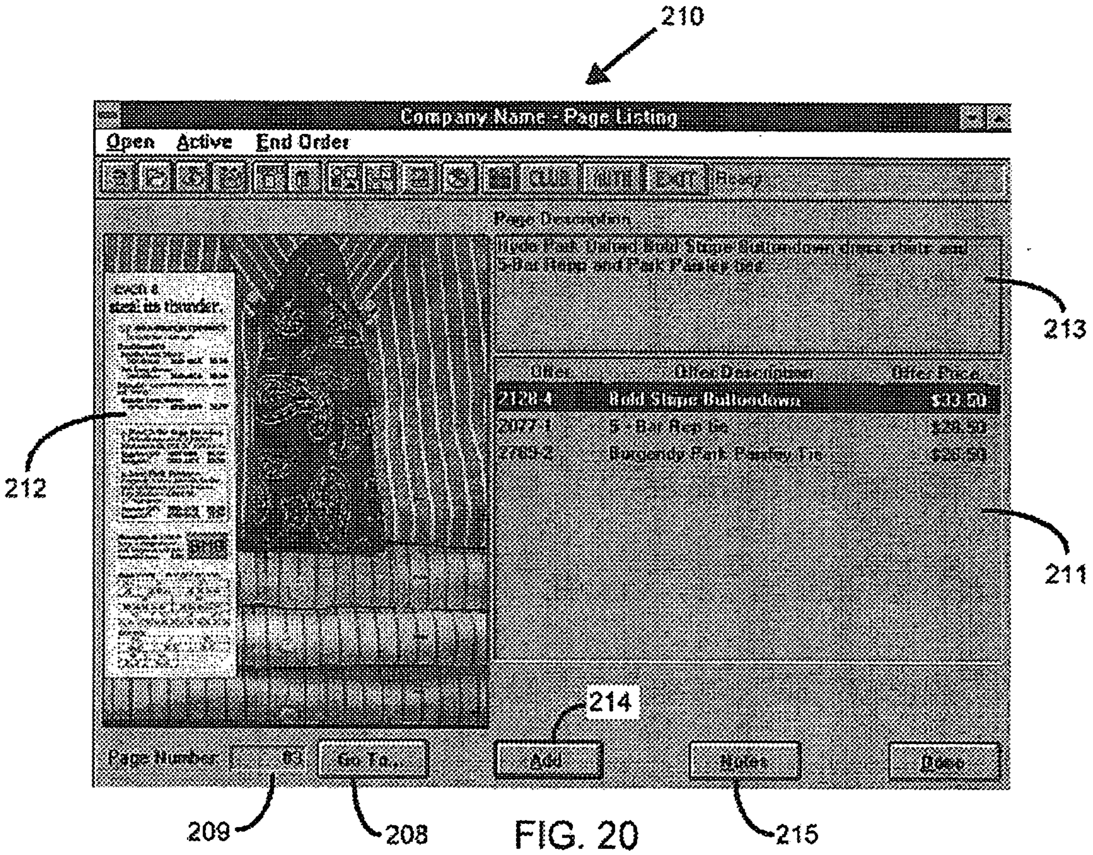

190
Company Name - Page Search
Open
Active
End Order
EDIT
EXIT
Source Search
194
New Shapes April 1994
Page 82
Page 83
193
LANDS' END
DIRECT MERCHANTS
192
Issue Date 3/31/94
198
Source
Done
First
Previous
Display Page
83
Next
Last
195
Search Offers
Go to Page
200
191
196

FIG. 19



Page Description

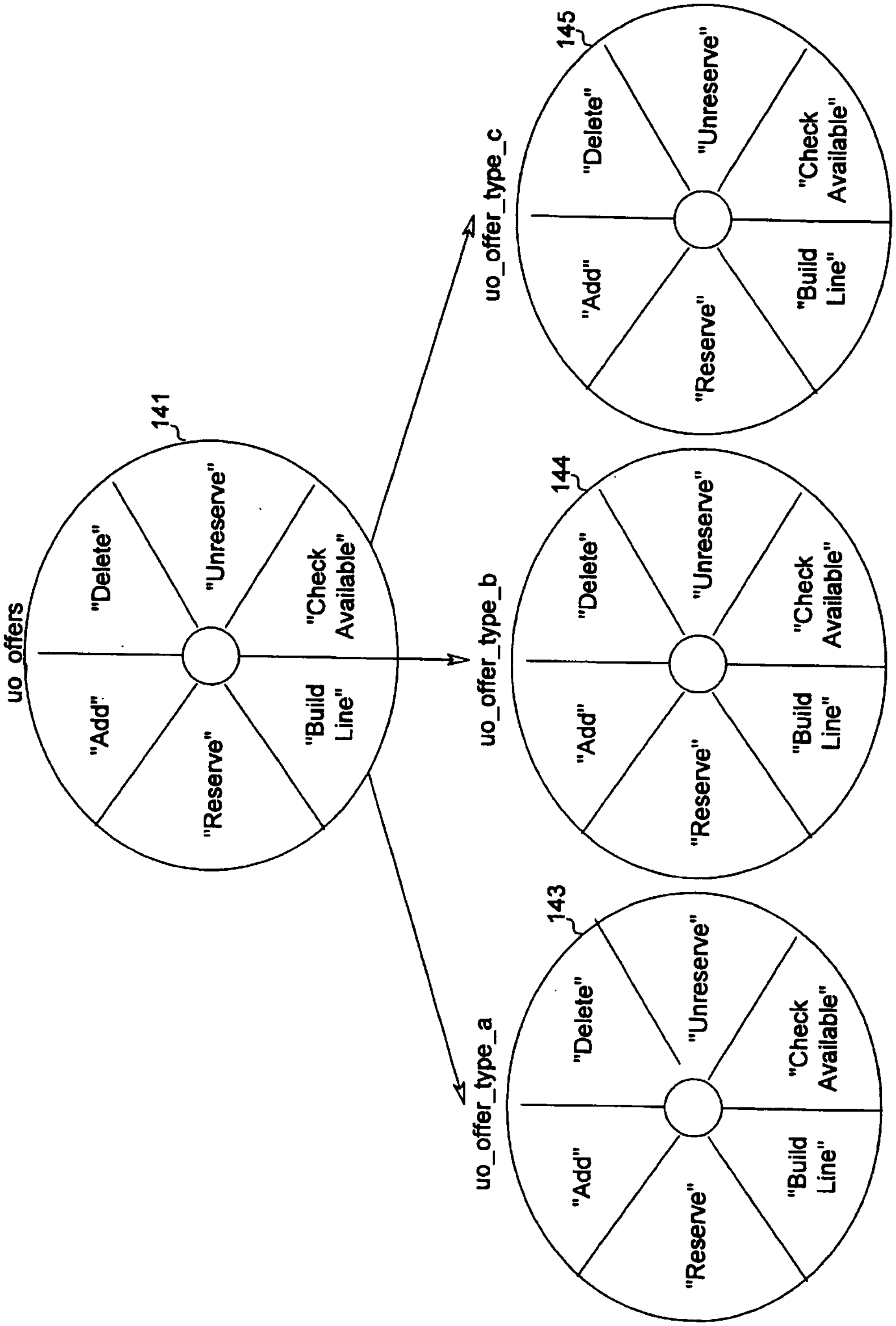
Hyde Park Oxford Bold Stripe Buttondown dress shirts and 5-Bar Repp and Park Paisley ties.

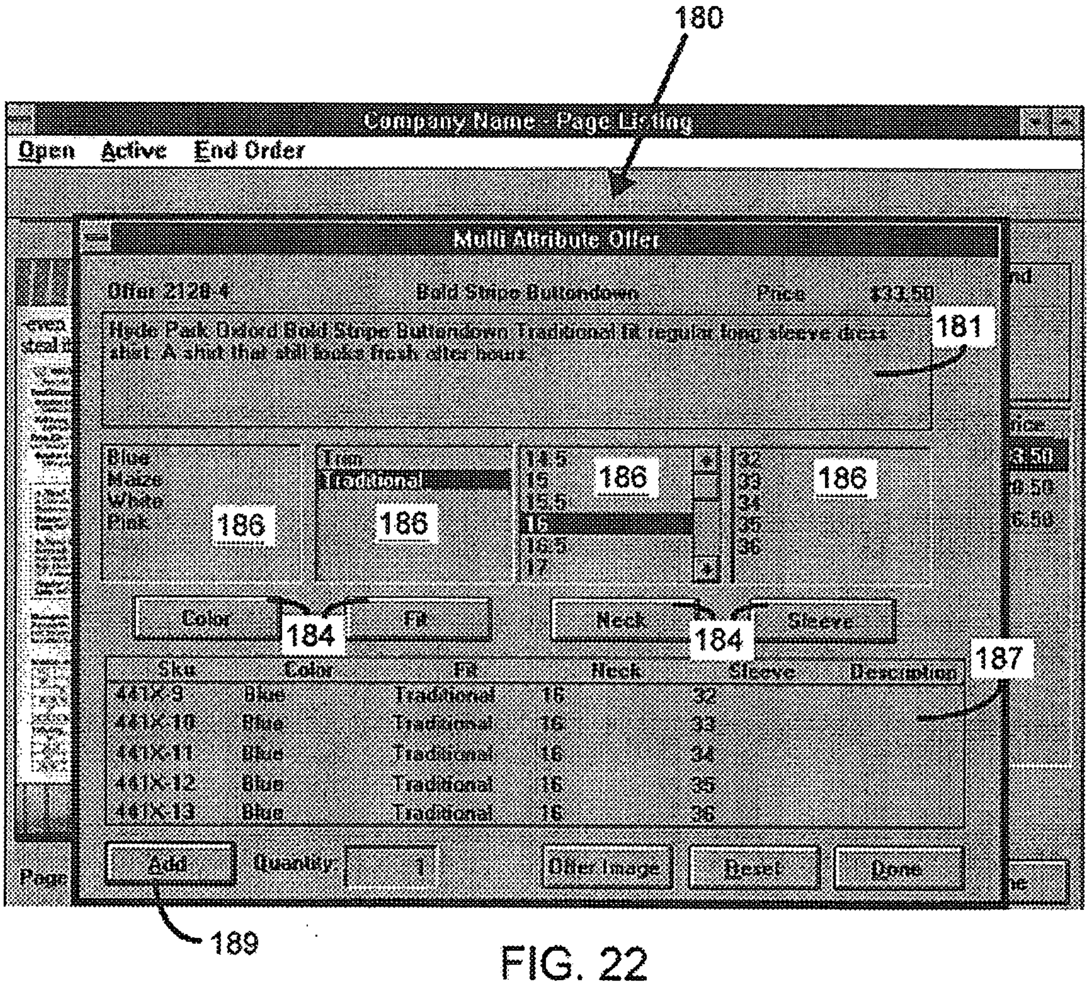
Offer	Offer Description	Offer Price
2128-4	Bold Stripe Buttondown	$33.50
2077-1	5 - Bar Rep Tie	$28.50
2789-2	Burgundy Park Paisley Tie	$28.50

Page Number 63

FIG. 20

FIG. 21


uo_offers
141
"Add"
"Delete"
"Unreserve"
"Check Available"
"Build Line"
"Reserve"
uo_offer_type_a
143
uo_offer_type_b
144
uo_offer_type_c
145



Offer 2128-4 Bold Stripe Buttondown Price $33.50

Hyde Park Oxford Bold Stripe Buttondown Traditional fit regular long sleeve dress shirt. A shirt that still looks fresh after hours.

Blue
Maize
White
Pink

Trim
Traditional

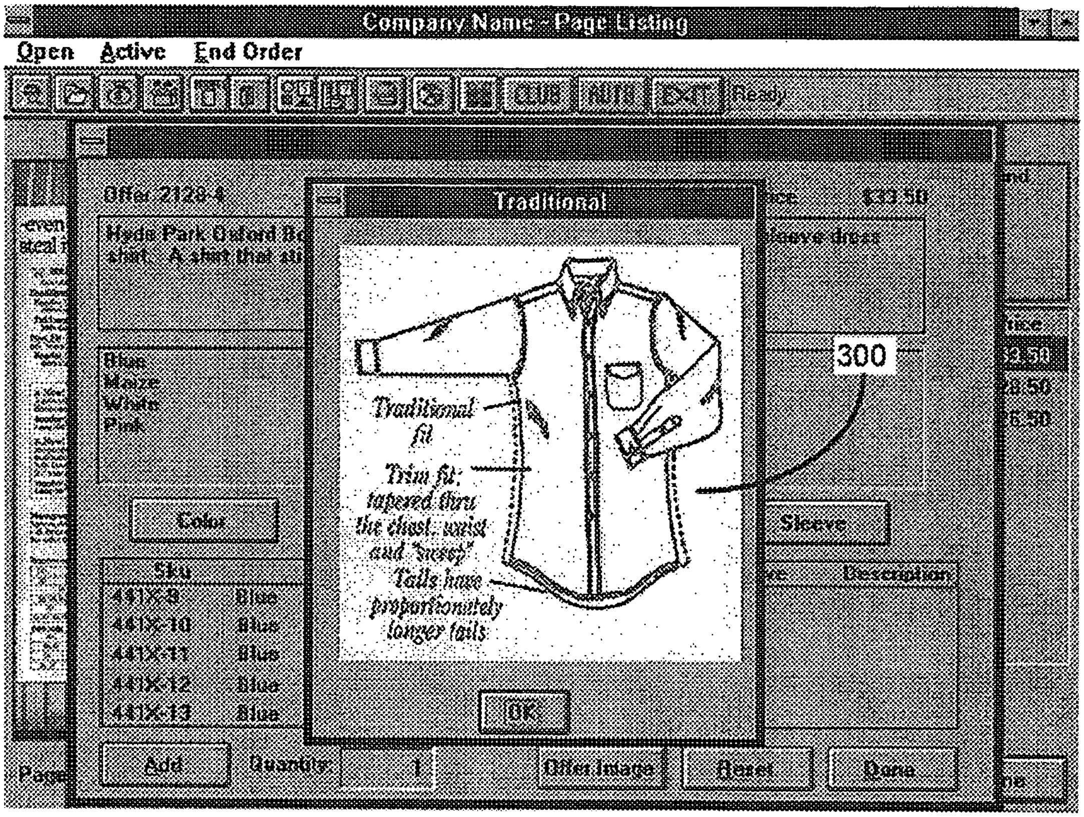
14.5
15
15.5
16
16.5
17

32
33
34
35
36

Sku	Color	Fit	Neck	Sleeve	Description
441X-9	Blue	Traditional	16	32	
441X-10	Blue	Traditional	16	33	
441X-11	Blue	Traditional	16	34	
441X-12	Blue	Traditional	16	35	
441X-13	Blue	Traditional	16	36	

Quantity: 1

FIG. 22


Company Name - Page Listing
Open
Active
End Order
CLUB
AUTO
EXIT
Ready
Traditional
Traditional fit
Trim fit: tapered thru the chest, waist and "sweep"
Tails have proportionately longer tails
300
OK
Color
Sleeve
Add
Quantity
Offer Image
Reset
Done
Page

Offer 2128-4

ice $33.50

Hyde Park Oxford Bo... sleeve dress

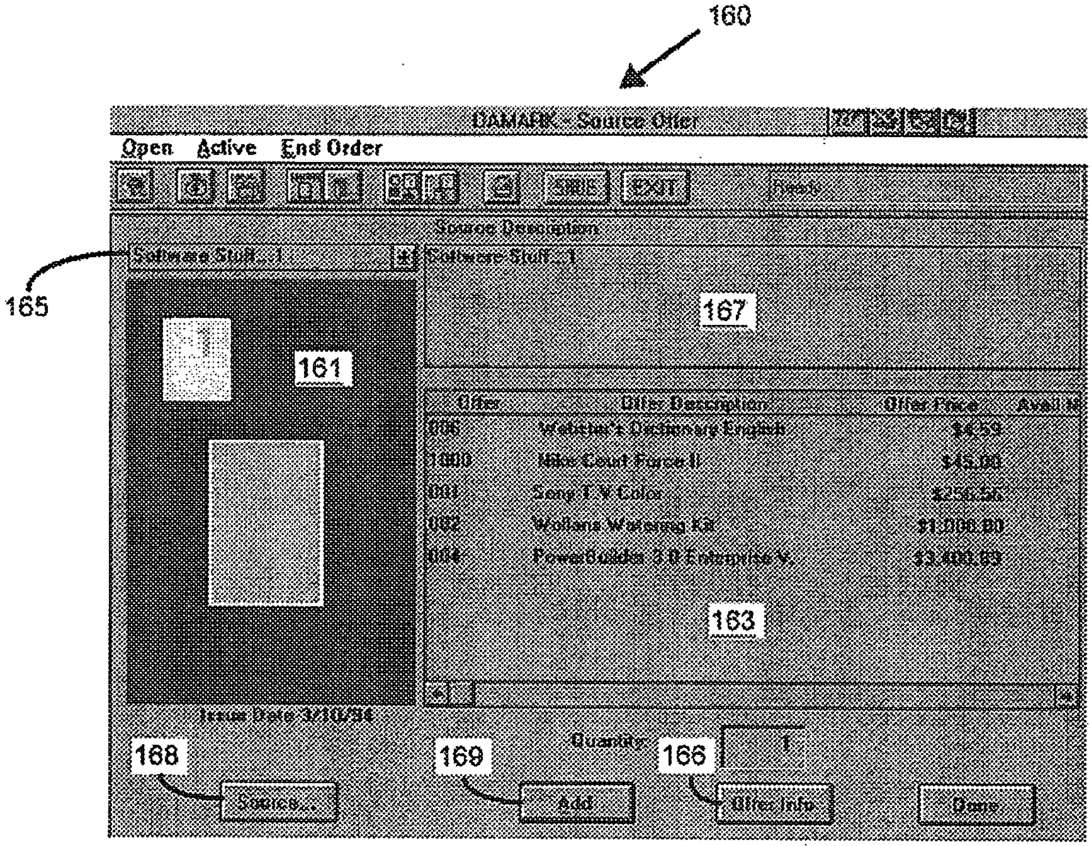
shirt. A shirt that sti...

Blue
Maize
White
Pink

Sku	
441X-9	Blue
441X-10	Blue
441X-11	Blue
441X-12	Blue
441X-13	Blue

Quantity: 1

FIG. 23



Source Description

Software Stuff...1

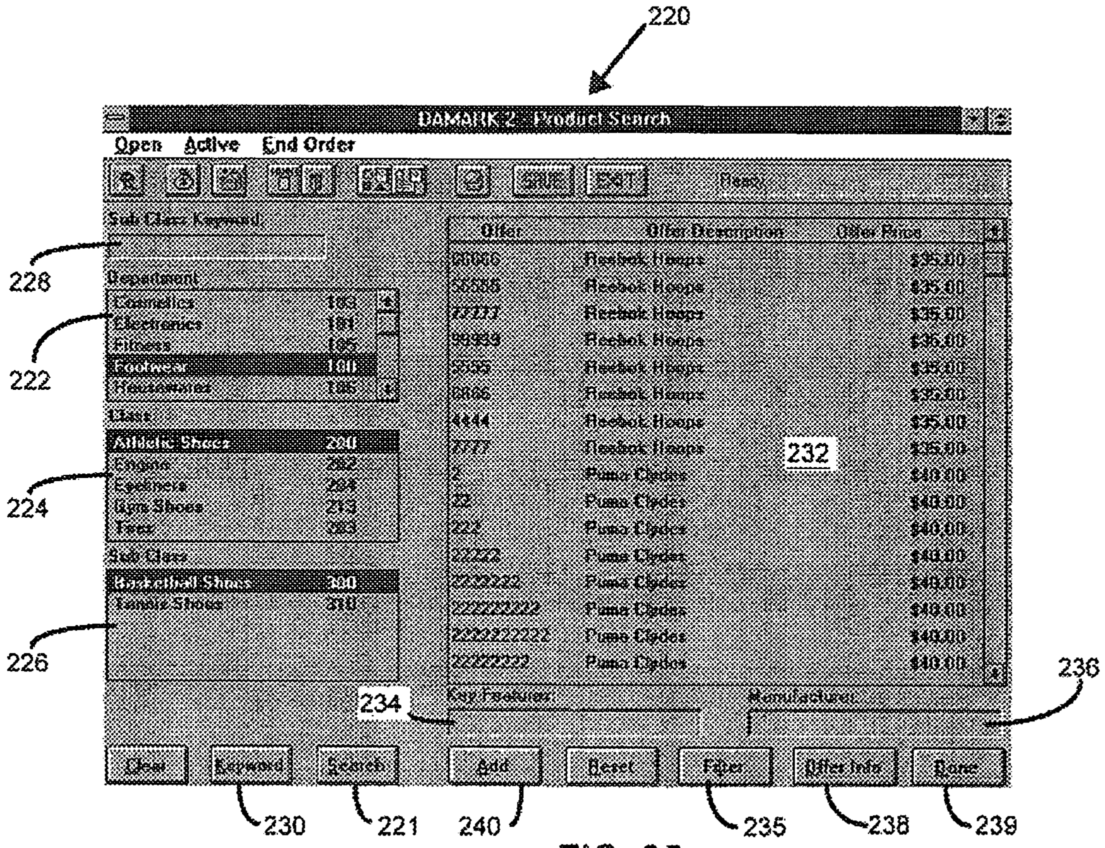
Software Stuff...1

Offer	Offer Description	Offer Price	Avail N
006	Webster's Dictionary English	$4.59	
1000	Nike Court Force II	$45.00	
001	Sony T.V Color	$256.56	
002	Wollans Watering Kit	$1,000.00	
004	PowerBuilder 3.0 Enterprise V.	$3,400.83	

Issue Date 3/10/94

Quantity: 1

FIG. 24



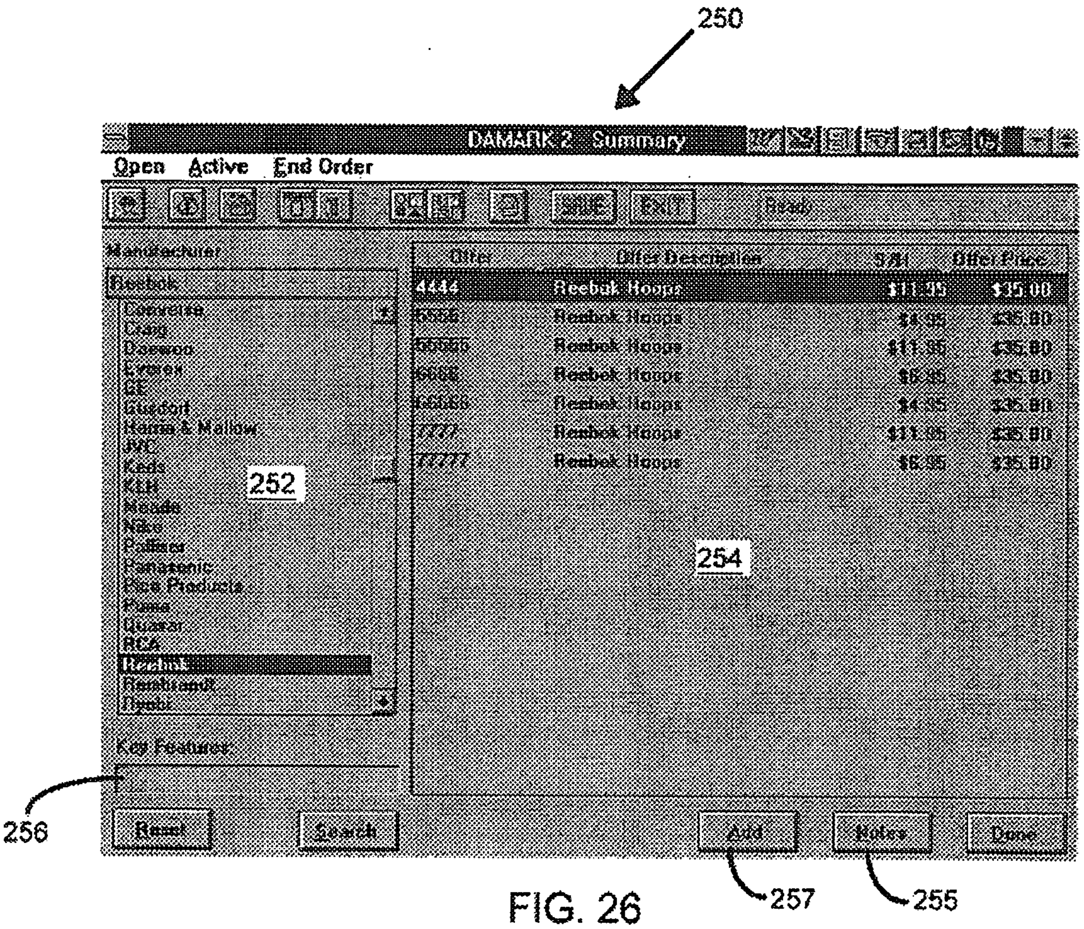
Department	
Cosmetics	183
Electronics	181
Fitness	185
Footwear	180
Housewares	186

Class	
Athletic Shoes	200
Engine	202
Eyeliners	204
Gym Shoes	213
Tires	203

Sub Class	
Basketball Shoes	300
Tennis Shoes	310

Offer	Offer Description	Offer Price
66666	Reebok Hoops	$35.00
55555	Reebok Hoops	$35.00
77777	Reebok Hoops	$35.00
99999	Reebok Hoops	$36.00
5555	Reebok Hoops	$35.00
6666	Reebok Hoops	$35.00
4444	Reebok Hoops	$35.00
7777	Reebok Hoops	$35.00
2	Puma Clydes	$40.00
22	Puma Clydes	$40.00
222	Puma Clydes	$40.00
22222	Puma Clydes	$40.00
2222222	Puma Clydes	$40.00
222222222	Puma Clydes	$40.00
2222222222	Puma Clydes	$40.00
22222222	Puma Clydes	$40.00

FIG. 25



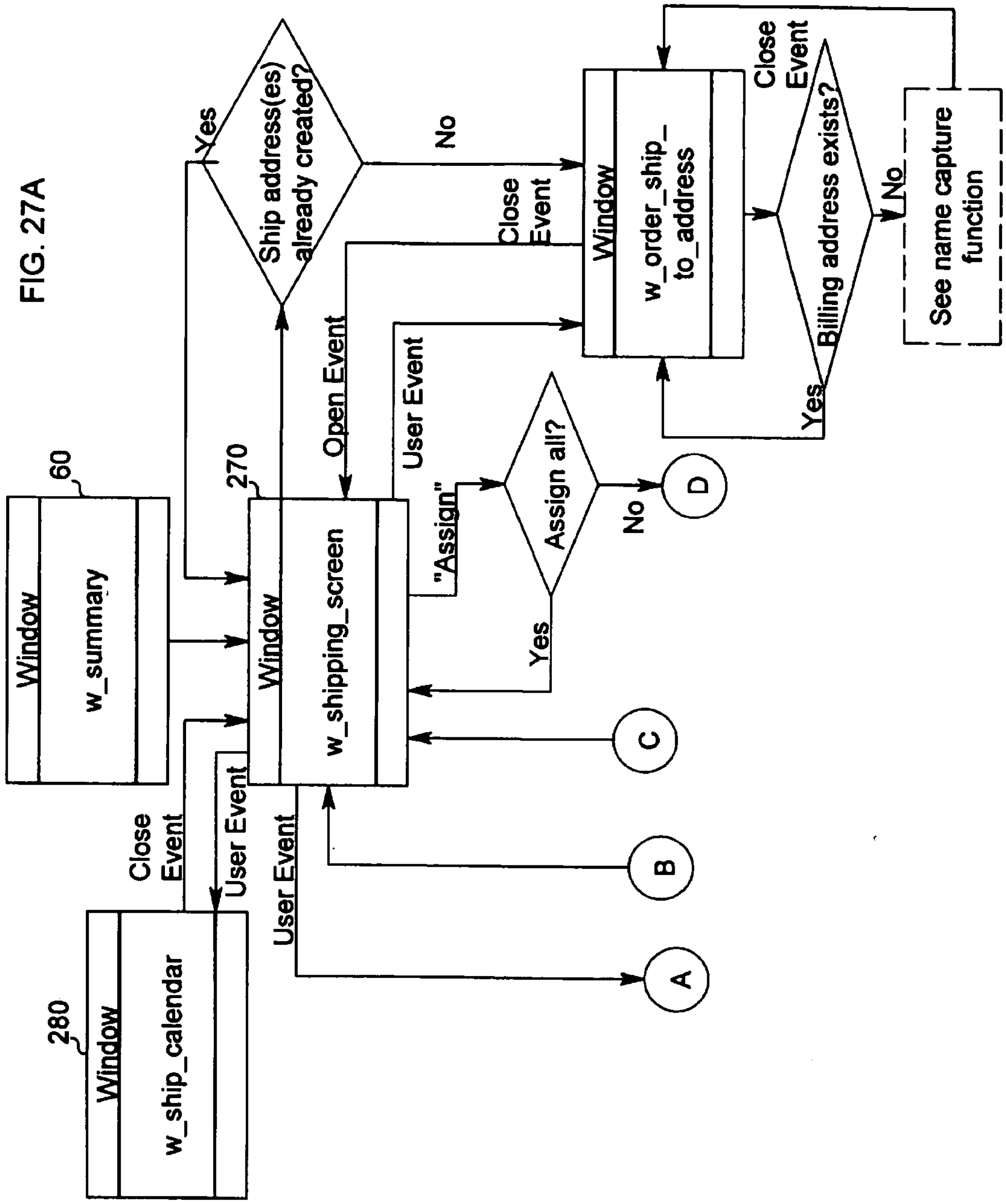
Reebok

Converse
Craig
Daewoo
Eveready
GE
Guidoff
Harris & Mallow
JVC
Keds
KLH
Meade
Nike
Palliser
Panasonic
Pico Products
Puma
Quasar
RCA
Reebok
Rembrandt
Ryobi

Offer	Offer Description	S/H	Offer Price
4444	Reebok Hoops	$11.95	$35.00
5555	Reebok Hoops	$4.95	$35.00
55555	Reebok Hoops	$11.95	$35.00
6666	Reebok Hoops	$6.95	$35.00
66666	Reebok Hoops	$4.95	$35.00
7777	Reebok Hoops	$11.95	$35.00
77777	Reebok Hoops	$6.95	$35.00

FIG. 26

FIG. 27A


Window
w_summary
60
Window
w_shipping_screen
270
Window
w_ship_calendar
280
Window
w_order_ship_to_address
Ship address(es) already created?
Yes
No
Open Event
User Event
Close Event
Close Event
Billing address exists?
Yes
No
See name capture function
"Assign"
Assign all?
Yes
No
D
C
B
A
Close Event
User Event
User Event

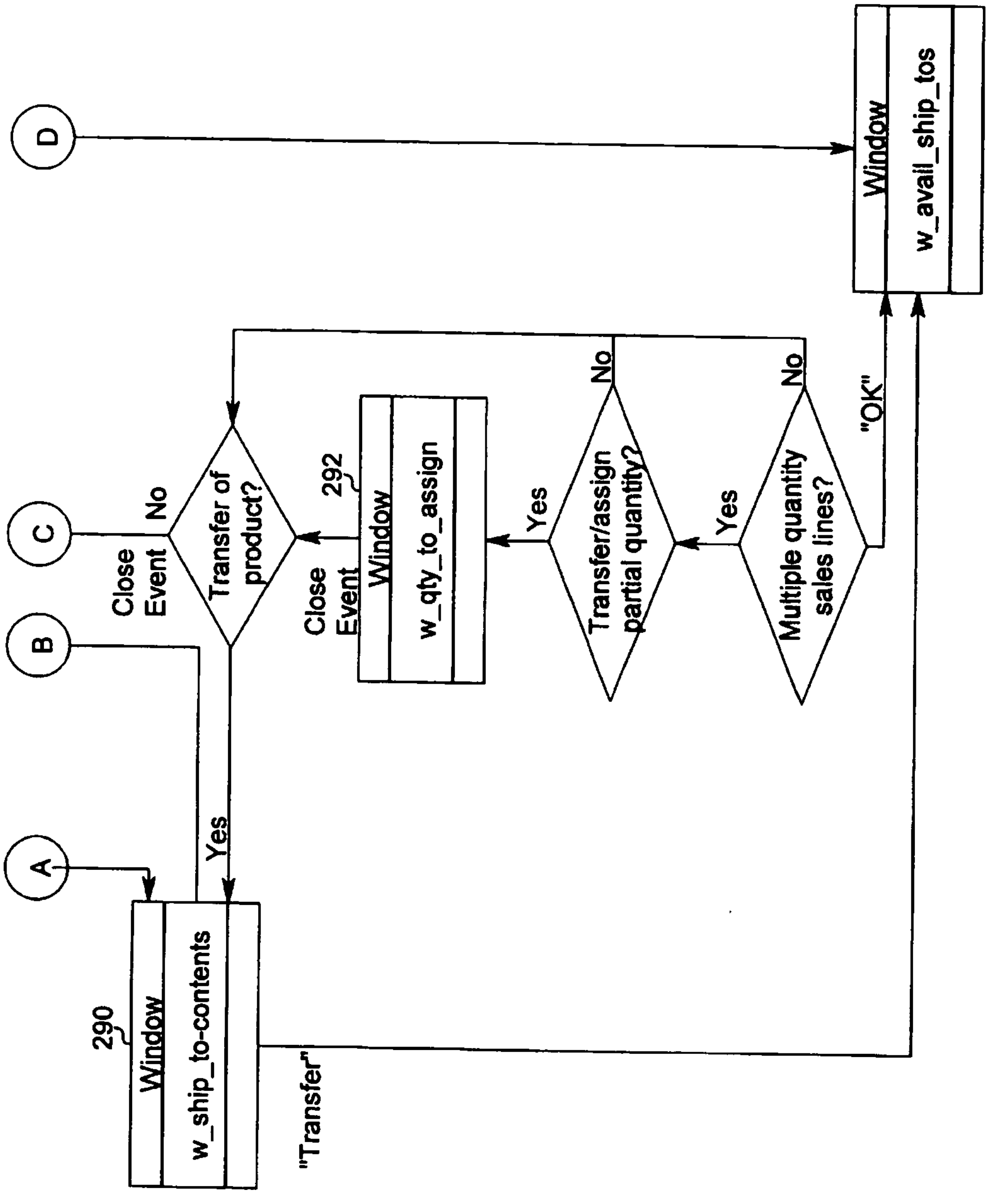
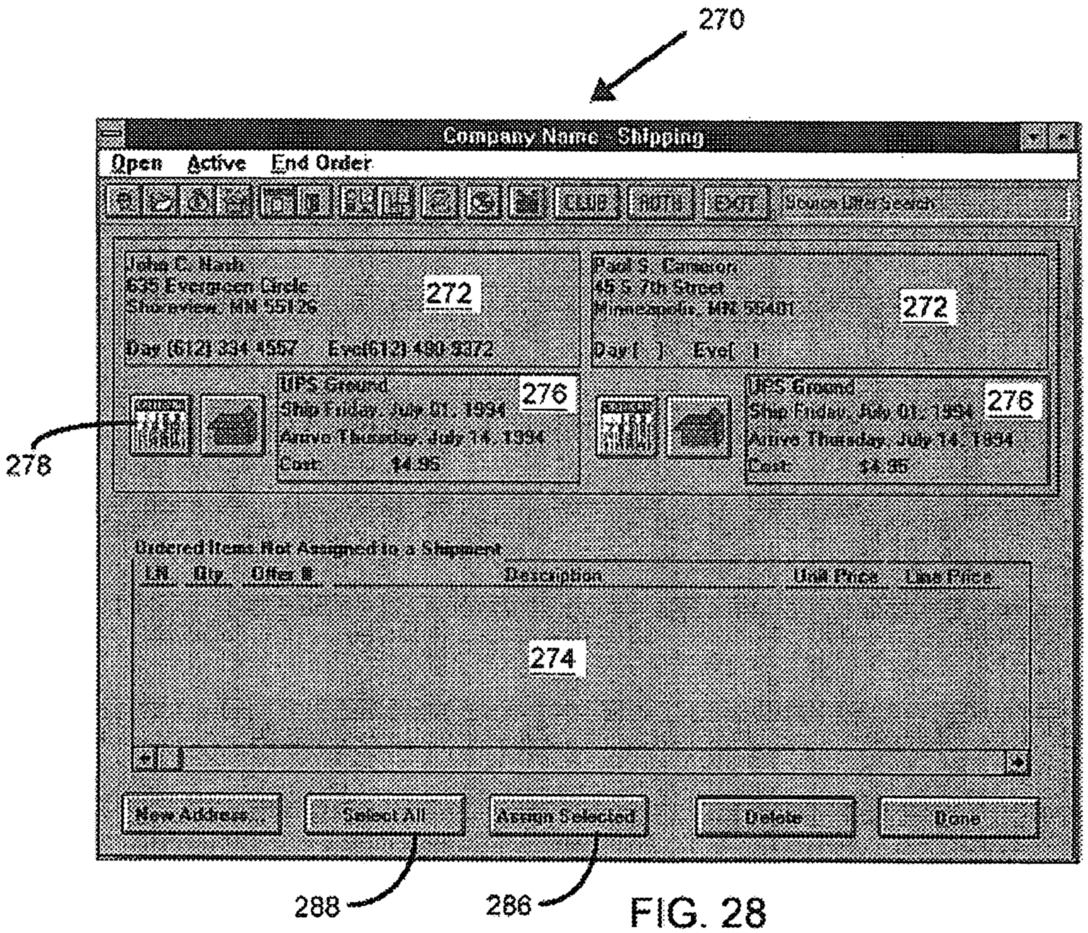

D
Window
w_avail_ship_tos
No
No
"OK"
Multiple quantity sales lines?
Yes
Transfer/assign partial quantity?
Yes
292
Window
w_qty_to_assign
Close Event
Transfer of product?
No
Close Event
C
B
A
Yes
290
Window
w_ship_to-contents
"Transfer"

FIG. 27B



John C. Nash
635 Evergreen Circle
Shoreview, MN 55126

Day (612) 334 4557 Eve(612) 490 8372

Paul S. Cameron
45 S. 7th Street
Minneapolis, MN 55401

Day ( ) Eve( )

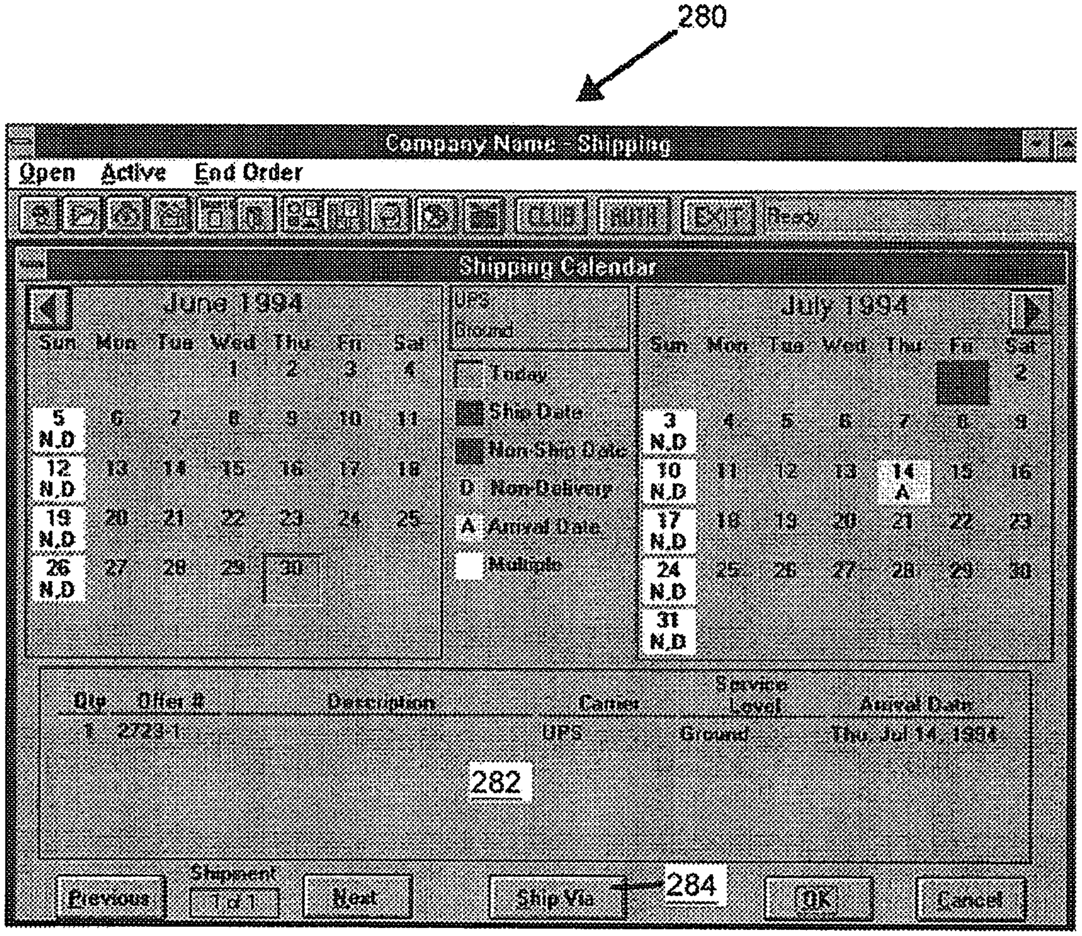
UPS Ground
Ship Friday, July 01, 1994
Arrive Thursday, July 14, 1994
Cost: $4.95

UPS Ground
Ship Friday, July 01, 1994
Arrive Thursday, July 14, 1994
Cost: $4.95

Ordered Items Not Assigned to a Shipment

LR	Qty	Offer #	Description	Unit Price	Line Price

FIG. 28



June 1994

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5 N,D	6	7	8	9	10	11
12 N,D	13	14	15	16	17	18
19 N,D	20	21	22	23	24	25
26 N,D	27	28	29	30		

UPS
Ground

□ Today
▨ Ship Date
▨ Non Ship Date
D Non-Delivery
A Arrival Date
□ Multiple

July 1994

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					▨	2
3 N,D	4	5	6	7	8	9
10 N,D	11	12	13	14 A	15	16
17 N,D	18	19	20	21	22	23
24 N,D	25	26	27	28	29	30
31 N,D						

Qty	Offer #	Description	Carrier	Service Level	Arrival Date
1	2723-1		UPS	Ground	Thu, Jul 14, 1994

Shipment
1 of 1

FIG. 29

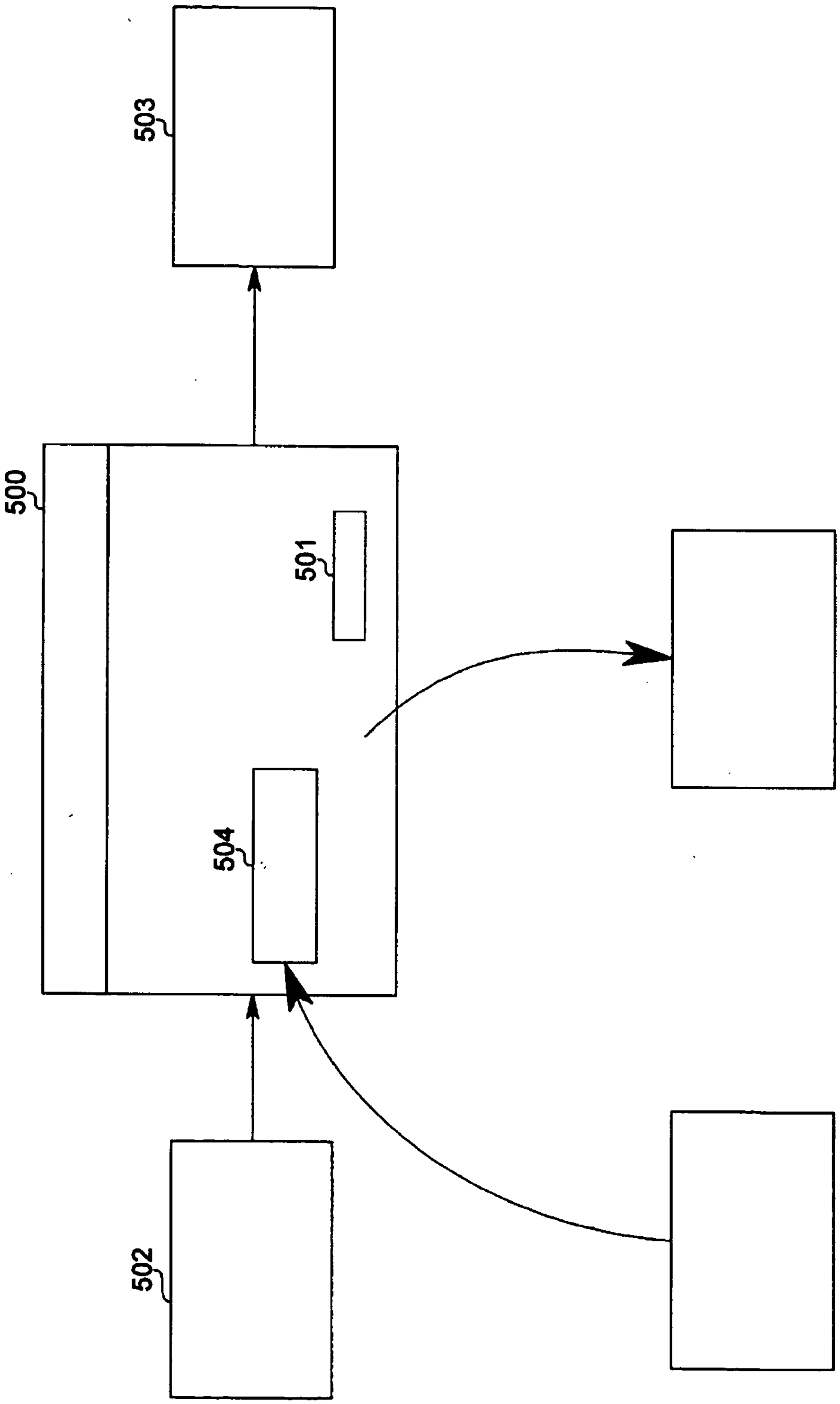
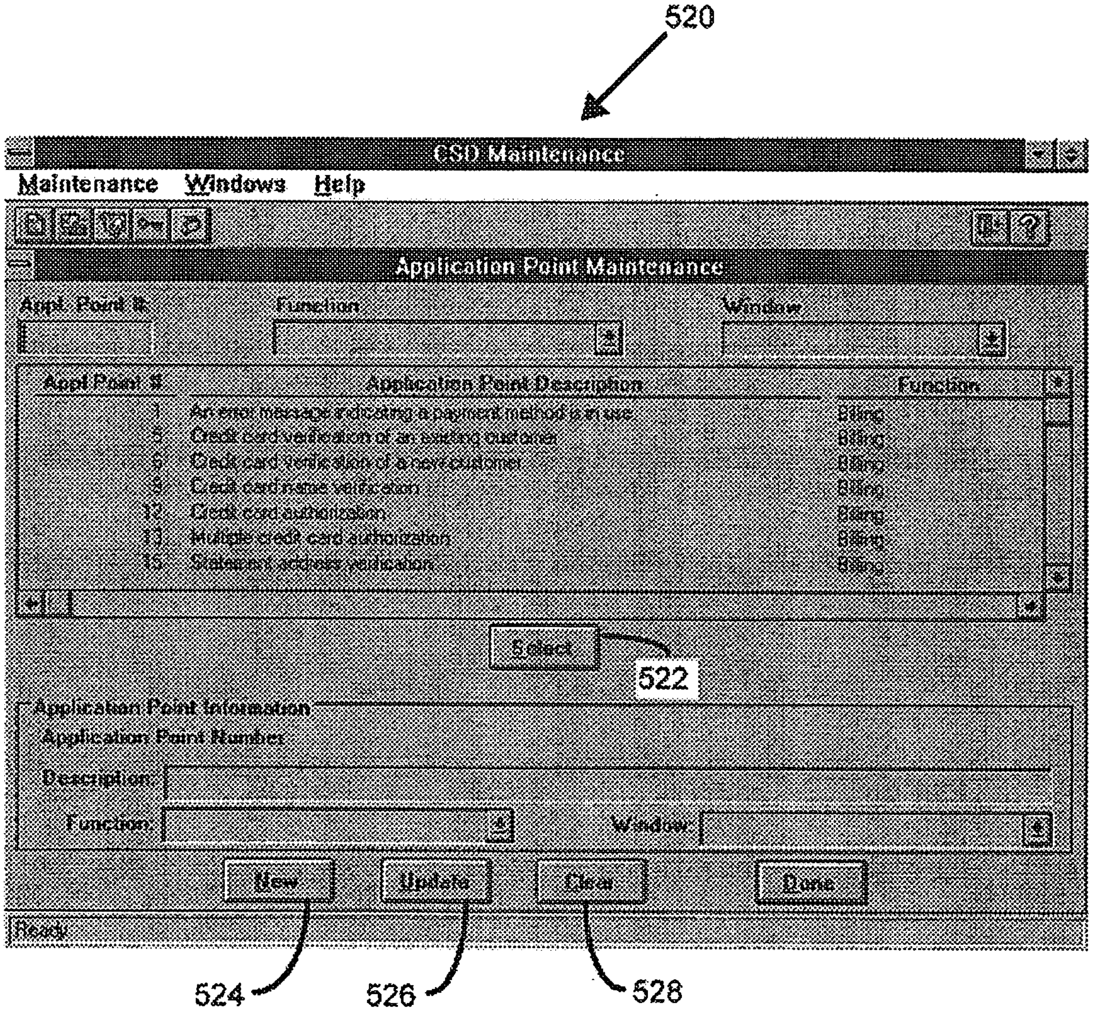
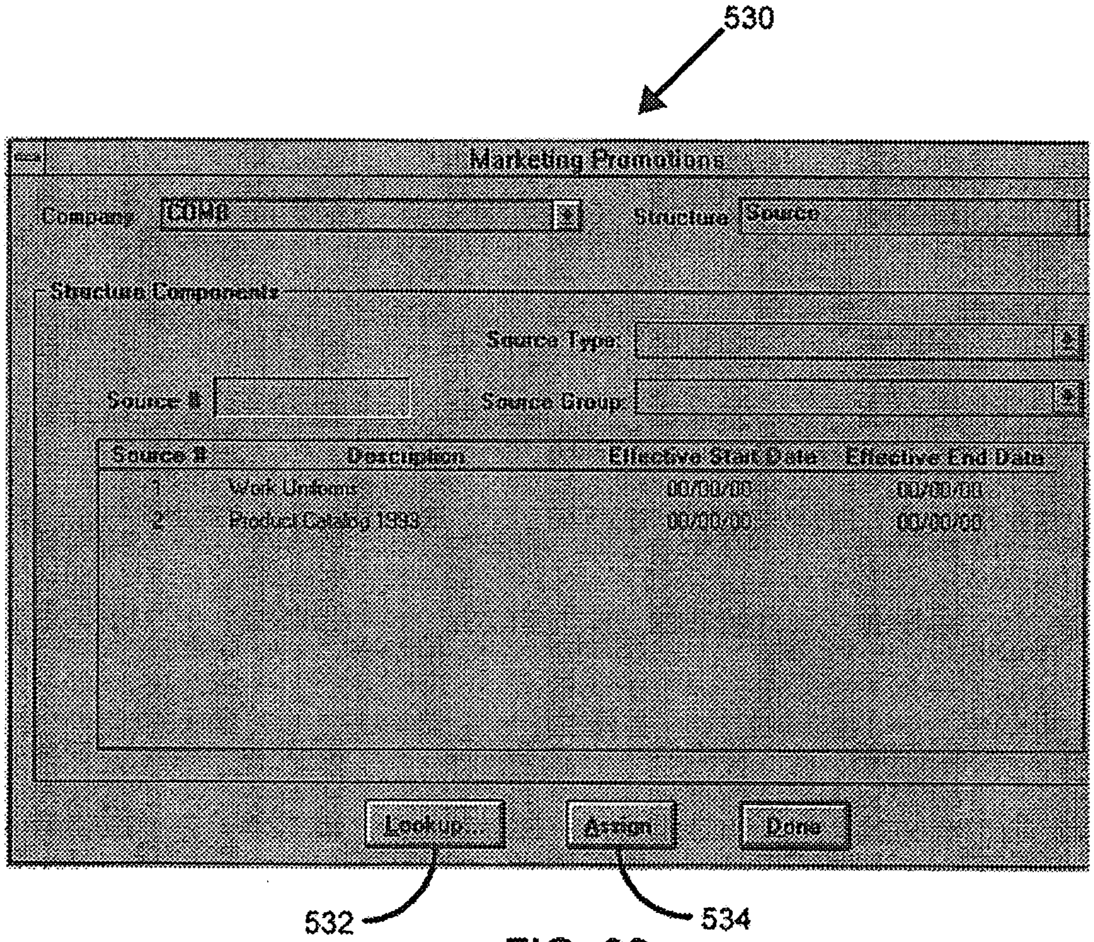


FIG. 30



Appl Point #	Application Point Description	Function
1	An error message indicating a payment method is in use	Billing
5	Credit card verification of an existing customer	Billing
6	Credit card verification of a new customer	Billing
8	Credit card name verification	Billing
12	Credit card authorization	Billing
13	Multiple credit card authorization	Billing
15	Statement address verification	Billing

FIG. 31



Company COMB
Structure Source

Structure Components

Source Type:

Source #

Source Group:

Source #	Description	Effective Start Date	Effective End Date
1	Work Uniforms	00/00/00	00/00/00
2	Product Catalog 1993	00/00/00	00/00/00

FIG. 32

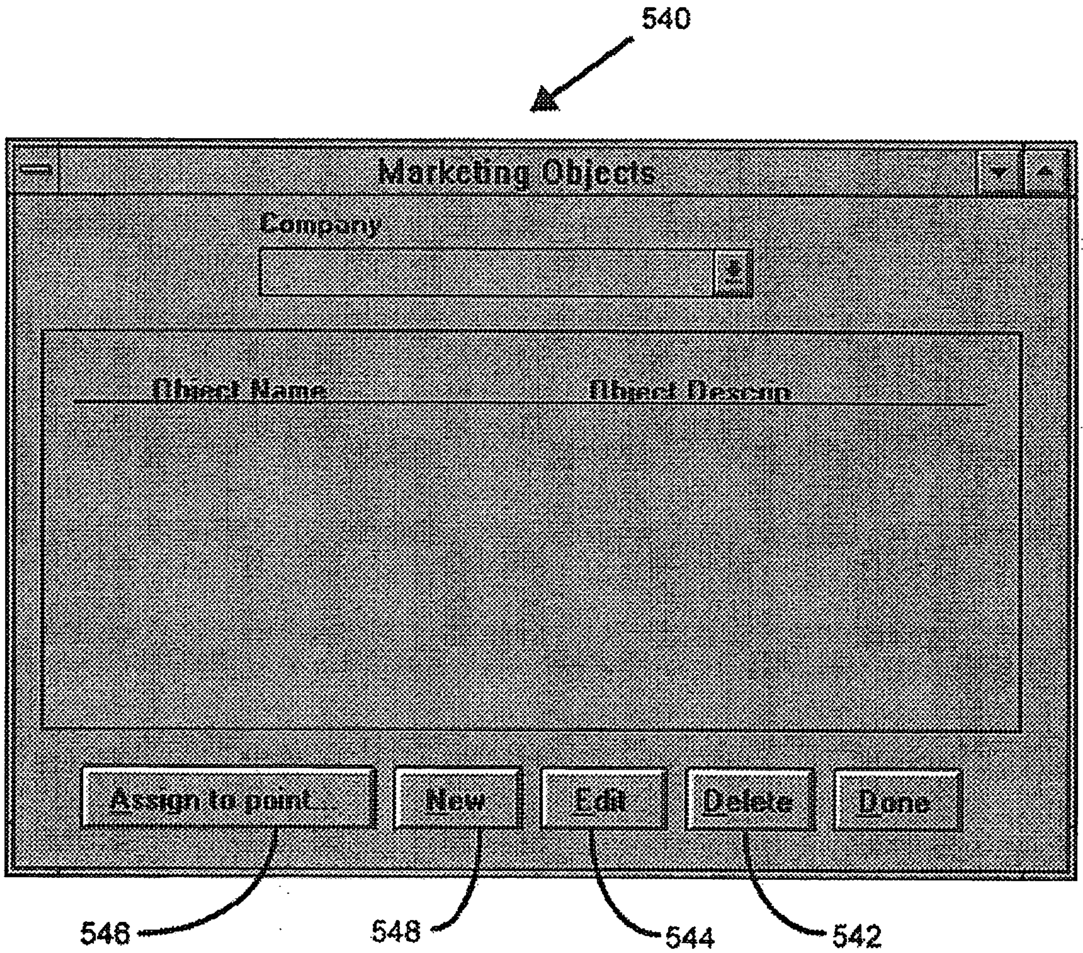
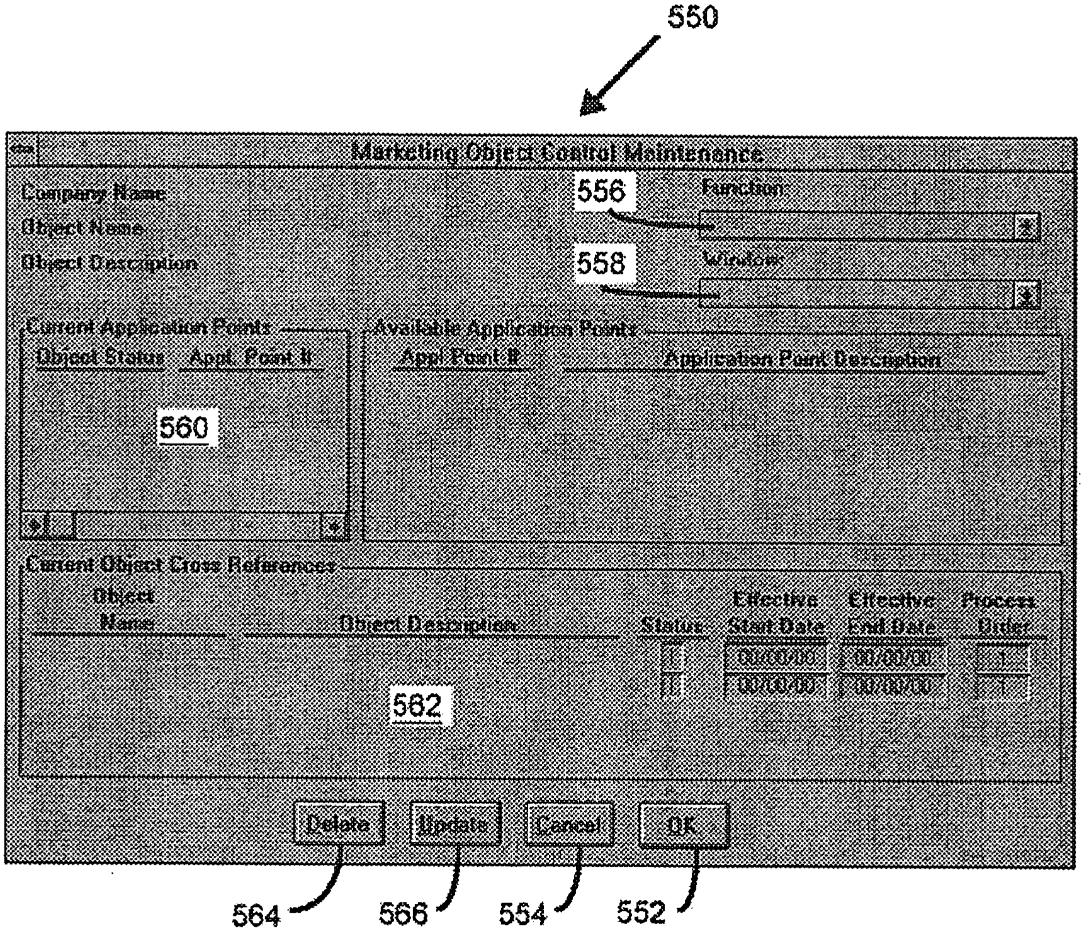
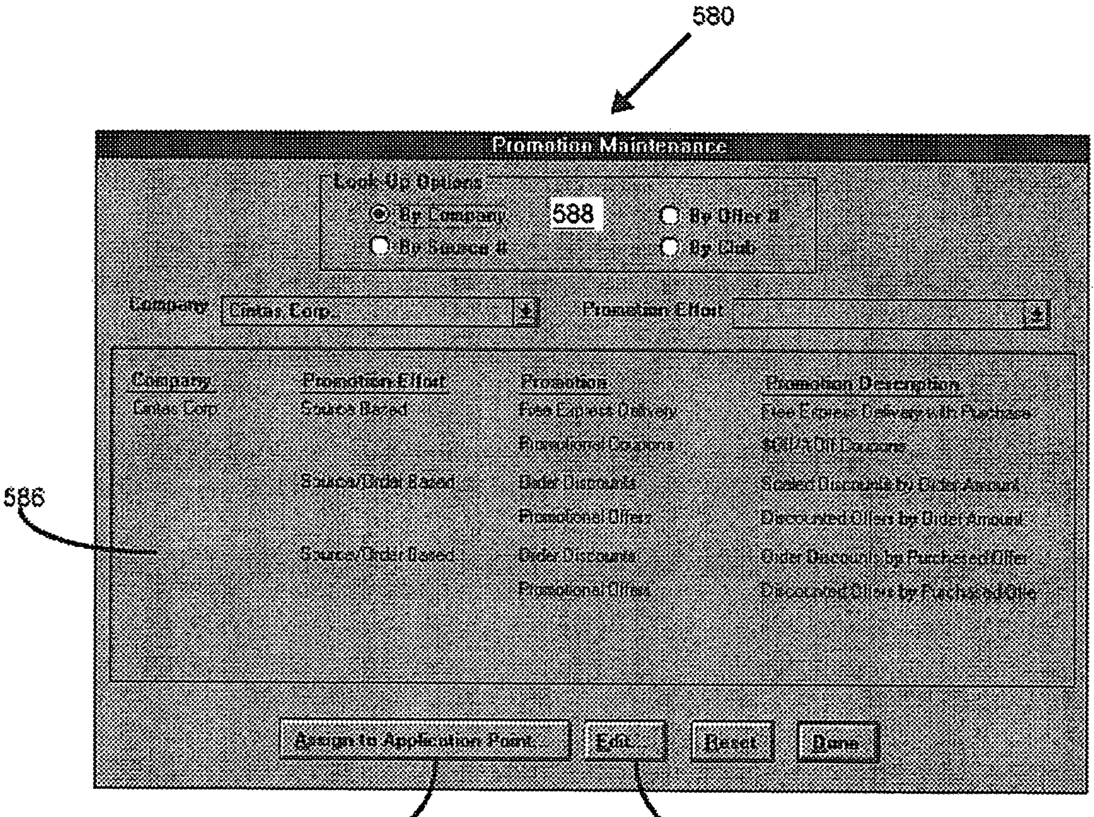

540
Marketing Objects
Company
Object Name
Object Descrip
Assign to point...
New
Edit
Delete
Done
546
548
544
542

FIG. 33


550
Marketing Object Control Maintenance
Company Name
Object Name
Object Description
556
Function
558
Window
Current Application Points
Object Status
Appl. Point #
560
Available Application Points
Appl Point #
Application Point Description
Current Object Cross References
562
Delete
Update
Cancel
OK
564
566
554
552

Object Name	Object Description	Status	Effective Start Date	Effective End Date	Process Order
		I	00/00/00	00/00/00	1
		I	00/00/00	00/00/00	1

FIG. 34



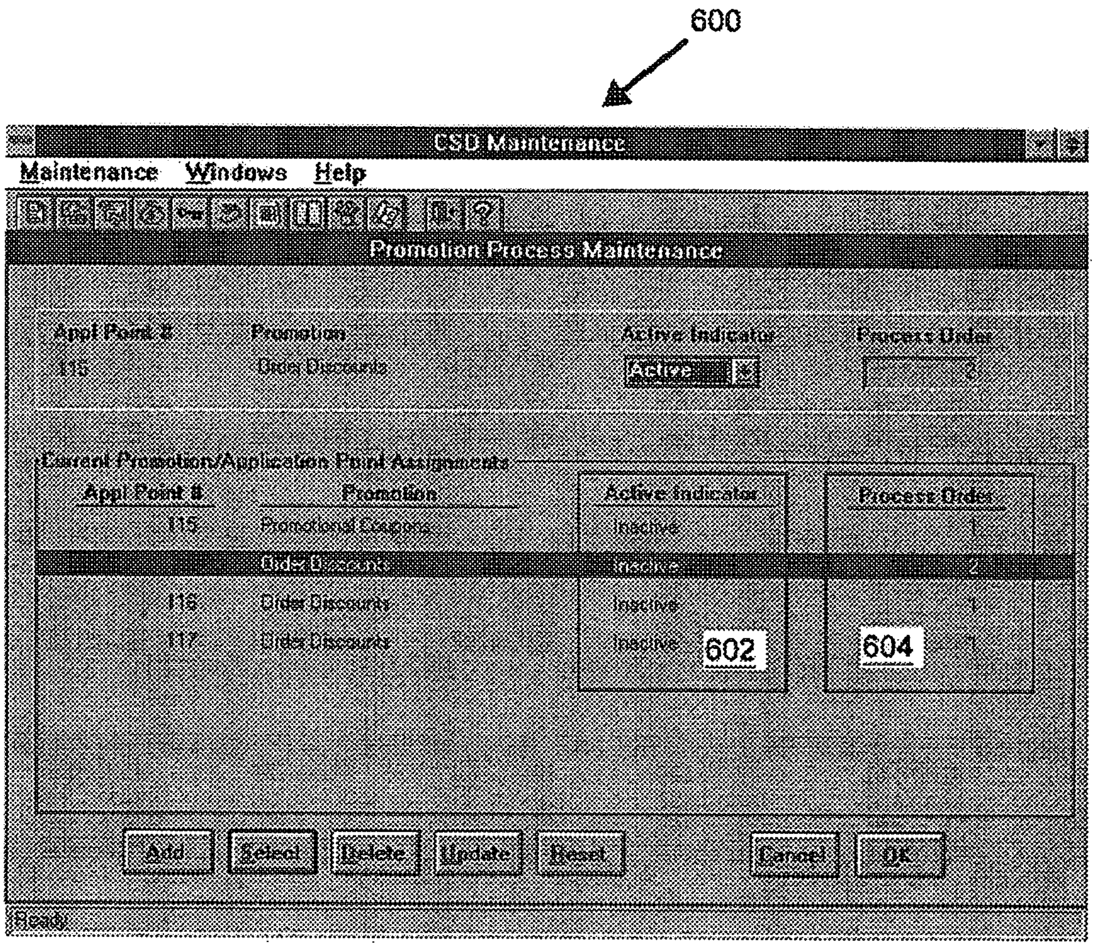
Promotion Maintenance

Look Up Options
● By Company
○ By Source #
○ By Offer #
○ By Club

Company: Cintas Corp.
Promotion Effort:

Company	Promotion Effort	Promotion	Promotion Description
Cintas Corp.	Source Based	Free Express Delivery	Free Express Delivery with Purchase
		Promotional Coupons	$10/$20 Off Coupons
	Source/Order Based	Order Discounts	Scaled Discounts by Order Amount
		Promotional Offers	Discounted Offers by Order Amount
	Source/Order Based	Order Discounts	Order Discounts by Purchased Offer
		Promotional Offers	Discounted Offers by Purchased Offer

FIG. 35



Appl Point #	Promotion	Active Indicator	Process Order
115	Order Discounts	Active	2

Current Promotion/Application Point Assignments

Appl Point #	Promotion	Active Indicator	Process Order
115	Promotional Coupons	Inactive	1
	Order Discounts	Inactive	2
116	Order Discounts	Inactive	1
117	Order Discounts	Inactive	1

FIG. 36

COMPUTERIZED SOURCE SEARCHING SYSTEM AND METHOD FOR USE IN AN ORDER ENTRY SYSTEM

This is a Divisional application Ser. No. 08/293,470, filed Aug. 19, 1994, now U.S. Pat. No. 5,592,378, which application(s) are incorporated herein by reference.

FIELD OF INVENTION

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyrights whatsoever.

The present invention relates to a computerized order entry system and method. The present invention is particularly useful in, but is not limited to, the telemarketing industry.

BACKGROUND OF THE INVENTION

Various types of structural complexes have been employed in the prior art by vendors of retailed products and services for storing products and for selling those products to consumers. Typically, the products are transported from the storage facility, such as a warehouse, to the store in which the products are sold. Consumers can purchase the products by going to the store during its business hours.

While this manner of selling products is commonplace, it has many disadvantages. One major disadvantage is cost. First, there are considerable start-up costs, such as the cost involved in furnishing the store. There is also the cost involved in maintaining the store. Typically, the vendor must not only provide the store itself, but also a parking lot capable of accommodating a reasonable number of vehicles of consumers who will visit the store. The rent on such a piece of property can be quite substantial, especially when the property is located in a major metropolitan area.

Due to the high costs of real estate, vendors are often forced to either locate their stores outside of metropolitan areas or to settle for smaller-sized facilities. In the former case, such locations are often less convenient for consumers. In the latter case, the vendor must sacrifice large product selection, thereby limiting the choice of products he or she may provide to the consumer.

Another problem associated with selling products through stores is theft. Vendors are forced to either absorb such losses or to install and maintain security systems, both of which further increase operating costs. Typically, these costs are passed on to the consumer through increased prices.

As a result of these disadvantages, it is becoming increasingly attractive to vendors to go directly to the consumer as a means to market their goods, such as through mail order, cable television, telemarketing and other direct response channels. There are many advantages to such direct marketing approaches. First, such channels are more convenient since they eliminate the need for the consumer to visit a store. The consumer need only fill out a form or pick up the phone to place his or her order. Second, such channels can be made available to the consumer 24 hours a day, seven days a week. Third, such channels have no geographic limits. These channels can be reached by the consumer from anywhere and everywhere.

To be truly effective, however, direct response channels must accommodate the taking and processing of orders in a fast, efficient and yet simple manner. In a typical telemarketing system, the time for an agent to process a customer's telephone call affects the overall performance of the system. For example, if the time for an agent to collect customer information is long, the business may have to employ more agents and subscribe to more telecommunications facilities (e.g., telephone lines) in order to be able to answer telephone calls from other customers in a timely manner. Moreover, if it takes the agent a substantial amount of time to place the order, the business may lose the customer.

The introduction of the computer has helped somewhat in this respect. Computerized shopping systems, however, are typically operated by users that are not computer-educated. As a result, the quality of the user interface is critical. The system must be designed to have a low learning curve.

The system must also be able to handle vast amounts of information. For example, when a customer places an order from a catalog, the agent typically keeps copies of recent catalogs and manually thumbs through them to look up and confirm the products requested by the customer. This procedure can be quite cumbersome. Moreover, since the customer is unable to see the physical product, it is critical to convey accurate product information to the customer.

The system also must be able to support changes due to functional modifications, enhancements and growth. The user interface typically used by computerized shopping systems, however, is embedded in the code of the program. As a result, the user is constrained by the system and not driven by the customer's needs and desires.

Accordingly, a need exists for a customer driven order entry system that permits placement of an order in a timely and efficient manner. There is also a need for an order entry system that provides for the normalization of data in order to limit redundancies of data, and for access to a variety of database management systems. There is a further need for an order entry system that supports enhanced marketing strategies and streamlined operations.

SUMMARY OF THE INVENTION

The present invention relates to a computerized source searching system and method for placing orders for offers that are located by searching through an electronic equivalent of the offer source used by the person placing the order.

In accordance with one embodiment of the invention, a computerized system for the placement of an order by a user for at least one offer is provided. The system includes a storage mechanism for storing offer information relating to the offers, and also for storing electronic reproductions of the offer sources. Each of the offer sources contains at least one offer. The system also includes source searching means for locating certain ones of the electronic reproductions of the offer sources. The source searching means includes a source display to display electronic offer source images, where the electronic offer source images represent identifying portions of the offer sources. The source searching means also includes a source filter to receive source search criteria, and for causing the source display to display those of the electronic offer source images that meet the source search criteria. The source searching means further includes a source selector to allow selection of an electronic offer source image. The computerized system includes offer searching means to locate the offer associated with the selected electronic offer source image. The offer searching means includes a source segment display to display particular electronic images of a portion of the offer source represented by the offer source image, and further includes an offer selector to allow the offer associated with the displayed electronic images to be selected by the user.

In another embodiment of the invention, a computerized method for the placement of an order for an offer is provided. The method includes the step of storing offer information relating to the offers, and also for storing electronic reproductions of the offer sources. Each of the offer sources contains at least one offer. The method includes the step of locating electronic reproductions of the offer sources, which in turn includes the steps of displaying electronic offer source images, entering source search criteria, displaying the electronic offer source images that meet the source criteria, and selecting an electronic offer source image that represents an identifying portion of the desired offer source. The method further includes the step of displaying electronic images of part of the offer source represented by the selected electronic offer source image, and selecting one of the offers associated with the selected electronic image.

These and other features and advantages of the present invention will become readily apparent to those skilled in the art from the following detailed description and corresponding drawings. As will be realized, the invention is capable of modification without departing from the invention. Accordingly, the drawings and description are to be regarded as illustrative in nature, and not as restrictive.

BRIEF DESCRIPTION OF THE DRAWINGS

As illustrated in the accompanying drawings in which like reference characters generally refer to the same parts or elements throughout the views:

FIG. **1** is a perspective view of a computerized order entry system.

FIG. **2** is a business data model for the order entry system of FIG. **1**.

FIG. **3** is a preferred logical data model for the order entry system of FIG. **1**.

FIG. **4**A is a preferred logical data model of the company structure of FIG. **3**.

FIG. **4**B is a preferred logical data model of the customer structure of FIG. **3**.

FIG. **4**C is a preferred logical data model of the product structure of FIG. **3**.

FIG. **4**D is a preferred logical data model of the order structure of FIG. **3**.

FIGS. **5**A and **5**B are an interconnected block diagram used to illustrate the meaning of the symbols in the data models of FIGS. **4**A–**4**D.

FIG. **6** is a conceptual block diagram of a functional data model of the order entry system of FIG. **1**.

FIG. **7** is a preferred summary user interface for the order entry system of FIG. **1**.

FIG. **8** is a preferred user interface for the summary module of FIG. **6**.

FIG. **9** is a preferred detailed dialog flow for the summary module of FIG. **6**.

FIG. **10** is a preferred detail dialog flow for the name capture module of FIG. **6**.

FIG. **11** is a preferred name capture user interface for the order capture module of FIG. **6**.

FIG. **12** is a preferred detail dialog flow for the billing module of FIG. **6**.

FIG. **13** is a preferred order payment user interface for the billing module of FIG. **6**.

FIG. **14** is a preferred credit card entry user interface for the order entry system of FIG. **1**.

FIG. **15** is a preferred object inheritance structure for a payment object for the order capture module of FIG. **6**.

FIG. **16** is a preferred detailed window data flow for a user interface of the billing module of FIG. **6**.

FIG. **17** is a preferred detail dialog flow for the order capture module of FIG. **6**.

FIG. **18** is a preferred source search user interface for the order capture module of FIG. **6**.

FIG. **19** is a preferred page search user interface for the order capture module of FIG. **6**.

FIG. **20** is a preferred page offer user interface for the order capture module of FIG. **6**.

FIG. **21** is a preferred object inheritance structure for an offer object of the order capture module of FIG. **6**.

FIG. **22** is a preferred multi-attribute user interface for the order capture module of FIG. **6**.

FIG. **23** is a preferred attribute information user interface for the order capture module of FIG. **6**.

FIG. **24** is a preferred source offer user interface for the order capture module of FIG. **6**.

FIG. **25** is a preferred offer search user interface for the order capture module of FIG. **6**.

FIG. **26** is a preferred manufacturer search user interface for the order capture module of FIG. **6**.

FIG. **27** is a detailed dialog flow for the shipping module of FIG. **6**.

FIG. **28** is a preferred shipping user interface for the shipping module of FIG. **6**.

FIG. **29** is a preferred shipping calendar user interface for the shipping module of FIG. **6**.

FIG. **30** is a preferred marketing function of the order entry system of FIG. **1**.

FIG. **31** is a preferred application point maintenance user interface for the order entry system of FIG. **1**.

FIG. **32** is a preferred marketing promotion user interface for the order entry system of FIG. **1**.

FIG. **33** is a preferred marketing objects user interface for the order entry system of FIG. **1**.

FIG. **34** is a preferred marketing objects maintenance user interface for the order entry system of FIG. **1**.

FIG. **35** is a preferred promotion maintenance user interface for the order entry system of FIG. **1**.

FIG. **36** is a preferred promotion process maintenance for the order entry system of FIG. **1**.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the following detailed description of the preferred embodiments, reference is made to the accompanying drawings which form a part hereof and in which is shown by way of illustration of a specific embodiment in which the invention may be practiced. This embodiment is described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that structural or logical changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense and the scope of the present invention is defined by the appended claims.

FIG. **1** illustrates an order entry system **10**. The preferred network configuration of order entry system **10** includes a plurality of servers 6, data entry devices 7, back-end systems 8 and databases 9, and Program 25, which represents a computer program, or program code on a computer-readable memory for direction the actions of order entry system 10. Program 25 preferably resides in servers 6, but could alternatively reside in data entry 7. In a preferred form, servers 6 are Unix®-based, data entry devices 7 are personal computers running Microsoft® Windows software, back-end systems 8 include inventory control and distribution applications, and databases 9 are relational databases, such as Oracle® or Sybase®. The network configuration preferably supports the TCP/IP network protocol.

Data entry devices 7 can be any type of data entry device, such as computers, multimedia kiosks or interactive television. In addition, the number of servers 6 is not critical. However, in order to improve performance, system 10 preferably distributes data over multiple servers so as to minimize the contention on any single server. The architecture of the preferred system also supports high performance and high scalability by using database segmentation, data location transparency, and multi-threading strategies. Such an architecture is critical for rapidly expanding businesses with high seasonal peaks.

The preferred order entry system 10 uses traditional programming language with fourth generation language tools, both using object based mechanisms. Unlike process-driven languages, this type of language operates in an event-driven mode. The object based event driven nature of the application allows the system to map logic against the business rules that are processed when the user initiates an event (e.g. “clicking” on a button, exiting a specific data capture field, etc.).

In addition, system 10 preferably uses a graphical user interface (GUI) which provides an efficient interaction between the user and the system. GUI allows system 10 to react to each keystroke instead of having to wait for an entire screen to be filled. This allows the application to execute fewer instructions per interaction, and thereby respond to the user more quickly. This level of responsiveness both reduces errors and enhances user satisfaction. It also provides a customer driven flow whereby the flow is driven by customer requests, rather than system constraints.

Order entry system 10 is preferably an object oriented system. With object oriented systems, functions performed by the system are each represented by an object. An object is a software packet containing a collection of related data and methods for operating on that data. Each method is made available to other objects for the purpose of requesting services of that object. Each object includes a set of related sub-functions. Accordingly, each object is preferably arranged as a structured collection of sub-functions, while each function should be arranged as a structured collection of objects.

Inheritance is a feature of object-oriented systems through which a new object can absorb the properties of an existing object. As a result, new objects (e.g., functions) may be added with minimal changes to existing objects, thereby significantly reducing development time and maintaining a consistent user interface to the user. Maintenance is also easier through the use of inheritance. Inheritance also allows the use of common objects across applications. As a result, a system that is capable of evolving over time to meet changing needs can be achieved.

FIG. 2 illustrates conceptually a preferred business data model 5 for order entry system 10 of FIG. 1. Business data model 5 identifies the major components of an organization'business and the relationship between those components. As can be seen in FIG. 2, the major components include a company 1, products 2, customers 3 and orders 4. Company 1 is the business entity that provides products and/or services for sale at a given price. Products 2 are those items that are produced and/or sold by the company and offered for sale to the customer at a given price. Customers 3 are those entities that wish to purchase the products from the company at a given price. Orders 4 are the vehicle by which the company can accurately record a sales transaction from the company to the customer for specific products at specific prices.

The components identified in FIG. 2 are organized into logical groups that make up a logical data model. A data model is a lower-level model of the entities that are of interest to a business and the relationships between those entities. FIG. 3 shows a preferred logical data model 11 for order entry system.

The logical groups within logical data model 11 correspond to the components of business data model 5 of FIG. 2 and include a company structure 12, a product structure 14, a customer structure 16, and an order structure 18. Logical data model 11 is constructed in such a way as to support changes in relationships and data entities due to functional modifications or enhancements to order entry system 10.

For the purposes of discussion only, order entry system 10 will be described for the telemarketing industry in which a user of the system is taking an order over the phone from a customer for offers available through a catalog. It is to be understood, however, that this system may be applied to other entry devices, such as multimedia kiosks and interactive television. With such devices, the user and the customer may be one and the same person.

Each of the logical data models for the logical groups shown in FIG. 3 are further illustrated in FIGS. 4A through 4D. FIG. 4A shows the logical data model for company structure 12. FIG. 4B shows the logical data model for product structure 14. FIG. 4C shows the logical data model for customer structure 16. FIG. 4D shows the logical data model for order structure 18.

In the data flow diagrams shown in FIGS. 4A through 4D, the boxes represent entities that are associated with data tables in a database. The number in the box corresponds to the particular data table from which data related to that entity is stored and retrieved. Preferably, the same piece of information is not stored on more than one data table so as to eliminate data redundancy. With such a construction, several advantages are obtained, such as the reduction of errors, more consistent data, physical storage space savings, and the reduction of back-up storage requirements.

FIGS. 5A and 5B illustrate the meaning of the symbols used in the interconnections between the boxes of each of the logical data models shown in FIGS. 4A through 4D. As shown in FIG. 5A, for an element 20, there must be at least one element 21 and there may be many elements 21. Each element 21 may be linked to one and only one element 20. As shown in FIG. SB, there may be zero or many elements 23 for each element 22. Each element 22 may be linked to zero or one element 23. Each of the logical data models shown in FIGS. 4A through 4D will now be discussed below.

FIG. 4A illustrates the relationships of the data tables in company structure 12. Company structure 12 consists of entities whose attributes are linked to a company. It is made up of information such as a company identification number and company name, as well as system parameters relating to company-specific functionality. Also included in company structure 12 are entities that present company-specific options within the application, such as types of payment methods accepted by the company and the carriers available for customer shipments.

FIG. 4B illustrates the relationships of the data tables in product structure 14. Product structure 14 consists of entities whose attributes are linked to the products that are offered for sale. It includes data relating to a catalog or the source of the product information, as well as the product offers that may contain one or many product stock keeping units (SKUs). In addition to the product structure attributes, product structure 14 also contains information about product pricing and product availability.

FIG. 4C illustrates the relationships of the data tables in customer structure 16. Customer structure 16 consists of entities whose attributes are linked to a company's customers. Customer structure 16 includes data relating to a customer or a customer's previous contact with the company. A customer identification number and address information, as well as the customer's phone numbers and previous payment methods are included in customer structure 16.

FIG. 4D illustrates the relationships of the data tables in order structure 18. Order structure 18 consists of entities whose attributes are linked to an order for specific products offered by a company and sold to a customer. It includes data relating to a specific sales transaction. Order structure 18 is made up of information such as order line data, an order's shipment method or methods, billing information, and order-specific customization (e.g., monogramming).

From business data model 5 of FIG. 2, several business functions may be identified. The functions performed by order entry system 10 are grouped into modules that make up the functional business model 24 of FIG. 6. The preferred modules include a summary module 30, a name capture module 32, an order capture module 34, a billing module 36, and a shipping module 38. Each module contains a number of conversations, which are the vehicle used to maintain the data tables in the corresponding data model. Preferred conversations for each module are conveniently listed in the description of the drawings and explained later.

The user interacts with order entry system 10 through a user interface. A user interface is something which bridges the gap between a user who seeks to control a device and the software and/or hardware that actually controls that device. The user interface for a computer is typically a software program running on the computer's central processing unit which responds to certain user-entered commands. Order entry system 10 uses object-based windows as the preferred user interface. In a preferred form, PowerBuilder® by Powersoft Corporation is used as the window development tool.

One preferred user interface is shown in FIG. 7. The user interface employs a window 15 preferably in the form of a rectangular shaped box having a toolbar 52 across the top which provides a set of standard menu options represented by a plurality of buttons 54*a* through 54*k*. Button 54*a* corresponds to a name capture function. Button 54*b* corresponds to a customer profile. Button 54*c* corresponds to a billing function. Button 54d corresponds to a shipping function. Button 54*e* corresponds to a source search function. Button 54*f* corresponds to a page search function. Button 54*g* corresponds to an offer lookup function. Button 54*h* corresponds to a manufacturer lookup function. Button 54*i* corresponds to a summary function. Button 54*j* corresponds to a discounts function. Button 54*k* corresponds to a gift card function.

Window 15 also includes a plurality of other buttons represented generally as 54. Buttons 54 typically represent an action or choice which is activated immediately upon user selection thereof. The buttons on window 15 may contain text, graphics or both. In a preferred form, buttons 54*a* through 54*k* contain graphics (i.e., icons) so that the user may readily determine the function they represent.

Window 15 preferably includes a plurality of data capture fields 17 for capturing data. The data capture fields allow the capture of variable length text. The data can be captured either automatically by system 10 or by the user, such as through a keyboard.

Each of the modules identified in FIG. 6 and their preferred conversations will now be discussed in detail below.

SUMMARY MODULE

Summary module 30 of FIG. 6 captures order information through window events and displays order information as it is completed. Summary module 30 is typically represented by summary window 60 as is shown in FIG. 8. The user may take an order via the toolbar 52 or by directly capturing information through the data capture fields on summary window 60 (i.e. the "fast path" method). The fast path method allows the user to enter an order when the customer possesses all of the necessary offer, billing and shipping information. It allows the user to perform all of the menu options represented by buttons 54*a* through 54*k* without having to activate each button separately. As a result, system 10 allows the user to navigate through all of the order entry functions associated with these buttons entirely from within summary window 60. Both methods of order taking, however, allow the taking of the order to be customer driven.

A detailed dialog flow for summary module 30 is shown in FIG. 9. With reference to FIGS. 8 and 9, the offer's source is entered in the source data capture field 62 of summary window 60. The source identifies in which catalog the customer found the offer. Upon capturing the offer number and offer quantity in the offer number data capture field 64 and quantity data capture field 65, respectively, an associated description, unit price and line price of the item is automatically displayed in data capture field 67 and inventory is automatically reserved.

Any of the offers listed in data capture field 67 may be added by selecting the add button 69. Offers may be deleted or quantities modified by selecting the delete button 66 and the update button 68, respectively. Offers having multiple attributes may be modified by selecting the modify button 70. Offers may be cancelled by selecting the cancel button 71. Offers that are comprised of other offers may be displayed via a popup/summary line detail window 61 by selecting the detail button 72. Window 61 lists offer information and the associated components of the offer, including a description of the component and the quantity of that component that is included in the order.

The billing address is preferably automatically captured at the bill to data capture field 74 when the user captures the customer number in the customer number data capture field 76. The preferred system verifies all customer numbers captured in customer data capture field 76 to determine whether they are a new, rented or existing customer.

A payment method may be captured when the user selects the payment method data capture field 78. Upon selection thereof, an order payment summary window 77 pops open where an existing payment method may be selected or a new payment method may be added. If a previous payment method is selected, the billing address is automatically captured into bill to data capture field 74. If a new payment method is selected, a credit card window 79 (see FIG. 14) pops open to capture the customer'credit information. Once the information is captured, a statement address window 81 may be opened to allow the user to verify the statement address. Financial billing information may be captured in data capture field 75.

The shipping address is captured into the ship to data capture field 80 by selecting the arrow button 82. Arrow button 82 also preferably automatically determines the best shipping carrier and service level for the offer being ordered by the customer. This determination is based, in part, on the shipment carriers and shipment service levels available, the offer being ordered and the shipment address. This feature is extremely important in light of the fact that point to point delivery providers are becoming more price competitive. The user, however, may override the system's determination if desired. This information, as well as the arrival date and shipping cost, are automatically displayed in the shipment data capture field 84.

Detailed information associated with a selected offer may be displayed via a product notes window 83 by selecting the notes button 50. Such information includes offer specifications and warranty information, for example.

A user may also preferably view all discounts that a customer receives for a particular order by selecting discounts button 54*j*. Upon selecting button 54*j*, a discounts window 73 opens. The information displayed by window 73 preferably includes the offer or offers being discounted, the type of discount, and the discount amount or amounts.

A user may also preferably view the customer's profile by selecting customer profile button 54*b*. Upon selection of button 54*b*, a customer profile window 15 as shown in FIG. 7 opens. The information displayed by window 73 preferably includes a textual and/or graphical representation of the customer's purchasing history, as well as a calendar for displaying the shipping status of any outstanding orders.

A user may also select a gift card by selecting gift card button 54*k*. Upon selection of gift card button 54*k*, a gift card window (not shown) opens, displaying at least one visual image depicting a representation of a gift card available through the system. In a preferred form, the gift card window displays a plurality of visual images depicting representations of a plurality of gift cards.

NAME CAPTURE MODULE

Name capture module 32 of FIG. 6 is used to identify the billing customer. A detailed dialog flow for this module is shown in FIG. 10. The preferred user interface for the name capture module 32 is represented by a name capture window 90 as is shown in FIG. 11. Name capture window 90 can be accessed by selecting name capture button 54*a* from toolbar 52 of the user interface of FIG. 7. Name capture window 90 defines a plurality of data capture fields into which customer information may be captured. Such information may include the customer's number, name, address, phone number and gender, to name a few.

Referring to both FIGS. 10 and 11, name capture window 90 may include an OK button 91, a cancel button 93, and a lookup button 95. OK button 91 validates the captured information and assigns a customer number if one does not exist. Cancel button 93 closes name capture window 90. Lookup button 95 opens a customer search criteria window 92 and is used to search for an existing customer, as will be further explained below.

When a user's phone rings, the preferred order entry system automatically determines the customer who is calling and the business or residence from which they are calling by using Automatic Number Identification/Dialed Number Identification (ANI/DNI). The initial data that displays in window 90 preferably is based on an ANI match or non-match. If a single ANI match is made, the data capture fields of window 90 may be automatically filled with the customer's information. If a multiple ANI match exists (i.e., more than one customer is listed for a designated phone number) a window preferably pops opens and displays the names and addresses of all customers who match the ANI number. If found, the user can select the correct customer from the list, and the appropriate data capture fields in name capture window 90 are automatically filled with the customer'information.

If there is not an ANI match, name capture window 90 is empty, ready to accept customer information from the user. The user can either elect to enter the customer's information directly into the appropriate data capture fields or to search for the customer. If a search is requested, customer search criteria window 92 pops open. Upon opening, the user may enter any combination of search criteria, such as, for example, the customer's name, postal code, phone number (s) or the customer number. If no existing customers match the search criteria, the user is returned to name capture window 90 with the search criteria information automatically captured into the appropriate data capture fields. The user can then directly capture the remaining information needed to process the order.

If one or more existing customers match the search criteria, a customer lookup select window 94 pops open with all matching customers retrieved therein. From customer lookup select window 94, the user may verify whether one of the customers retrieved represents the correct customer. If so, the information may be automatically captured into the appropriate data capture fields of name capture window 90. If not, system 10 preferably fills in the data capture fields corresponding to the search criteria. The user may then capture the remaining needed customer's information.

While the customer's information has been shown to be captured within name capture module 32, it can be understood that such information can also be captured in other modules, such as billing module 36.

BILLING MODULE

Billing module 36 is used to identify the payment method for the offer or offers being ordered by the customer. A detailed dialog flow for this module is shown in FIG. 12. Billing module 36 is typically represented by an order payment window 100, which is shown in FIG. 13. Order payment window 100 can be accessed by selecting the billing button 54c from toolbar 52 of the user interface shown in FIG. 7.

With reference to FIG. 13, order payment window 100 defines a customer payment method 102 in which the customer'previous payment methods may be displayed. For security reason, customer payment method data capture field preferably displays only proprietary in-house credit cards. The available payment methods for the company providing the offer or offers being ordered is displayed at available payment method data capture field 104. Such payment methods may include, for example, a credit card, a check, a coupon, and/or a recovery coupon (i.e., gift certificate). Order payment window 100 also displays any payment options at payment options data capture field 99. Examples of such payment options may include layaway, installments and the like.

Referring back to FIG. 12, from order payment window **100**, the user may select either an existing payment method or a new payment method. If a new payment method is selected, a payment method window **108** pops open. Payment method window **108** will change dynamically depending on which payment method type is selected (i.e., credit card window, coupon window, etc.). If payment method window **108** is opened as an existing payment method, the data capture fields defined therein are automatically filled in with the customer's payment history.

Upon selection of one of the payment methods, this history is automatically captured into selected payment method data capture field **106**. By eliminating manual re-entry of such information, errors in billing are reduced. A small visual image representation of the payment method is preferably included for each payment method listed in payment method data capture field **106** for easy verification that the selected payment method is correct. In this manner, errors in billing may be further reduced. If payment method window **108** is opened as a new payment method, the user must capture the payment information in the appropriate data capture fields.

An example of a payment method window **108** for a credit card payment is shown in FIG. **14**. Payment method window **108** of FIG. **14** defines a credit card number data capture field **115** and expiration date data capture field **117** into which the customer'credit card number and expiration date, respectively, may be captured. This window also preferably includes a visual image **111** of the selected credit card. This allows the user to easily verify the credit card being used by the customer and further helps to prevent errors in billing. The customer's billing statement address is captured at statement address data capture field **112**. In a preferred form, the customer's statement address will default to the address captured in bill to data capture field **74** of name capture window **90**.

The statement address captured in statement address data capture field **112** may be verified via a credit card address window **113** by selecting the address button **109** from payment method window **108** of FIG. **14**. The billing statement address, however, should not be changed by the user unless the customer has moved. If the billing statement address is different from the address given verbally by the customer, it may be a sign of credit card fraud. Therefore, if the user changes a billing statement address for an existing credit card, the preferred system will set a fraud status flag so that follow-up work can be performed to verify that the credit card was not stolen.

Once all of the information for the selected payment method has been captured, the user is returned to order payment window **100** with the selected payment method information automatically captured in selected payment method data capture field **106**. From there, the user may select to perform an electronic credit card authorization via an authorize window **114**. Once authorized, this payment method will be assigned to the order. In addition, if the payment is new, this payment method information will be captured into customer payment method data capture field **102**.

One of the key features of billing module **36** is the ability to allocate an order total across a plurality of payment methods. Any combination of the customer's previously used payment methods, or new payment method or methods may be assigned to an order as long as at least one payment method is selected. As is shown in FIG. 13, the customer may allocate either a dollar amount or a percent of the total order amount to each payment method, with the exception of coupons and gift certificates that state a specific dollar off amount. The dollar amount allocation is captured in dollar amount data capture field **116**, while the percent to allocate is captured in percent data capture field **118**. The amount to be billed is automatically calculated by the preferred order entry system and captured in the amount to be billed data capture field **110**. The total order amount is automatically calculated and captured in total order amount data capture field **105**, while the amount left to allocate is automatically calculated and captured in amount left to allocate data capture field **107**.

If a single payment method is chosen, one hundred percent (100%) of the order total is automatically allocated to that payment method. If more than one payment method is selected, the customer must choose how to allocate their payment methods. A recalculate button **101** is provided on billing window **100** which, when selected, calculates the dollar amount to be billed to each payment method based on the dollar amounts and percentages captured for each payment method. If the allocation is incomplete, the total amount of the order will be applied to the first payment method, less any coupon or gift certificate.

The allocation is preferably calculated in the following manner. First, each order payment method line is evaluated with the first payment method. Second, the percent-to-allocate for all rows is summed. This value normally must equal either one hundred percent (100 %) or zero percent (0%) (zero percent meaning that the order amount is fully allocated by dollar amount). If it does not, the user must either re-enter the allocation percent so that it equals one hundred percent (100%) or fully allocate the dollar amount of the order. When the first payment method is added to the order, the percent data capture field **118** preferably is automatically set to one hundred percent (100%) and will remain that way unless modified by the user. Third, the actual allocation calculations are performed. To begin with, all dollar amounts are allocated to each respective payment method. The allocation routine will allocate up to the current total order amount. If the dollar amount to allocate is greater than the total order amount, each row will be allocated beginning with the first row until the amount left to allocate is zero dollars ($0). The payment methods will not be over-allocated. If the total order amount is not completely allocated by dollar amount, the remaining amount to be allocated is calculated and allocated according to the percentages entered.

Other functions provided through order payment window **100** include deleting a selected payment method from the current order. A customer may wish to remove a selected payment method if they change their mind or if it is incorrect, such as if his or her credit card has expired. These functions may be accomplished via a delete button **120**.

Upon closing order payment window **100**, if the order is not completely allocated, the preferred order entry system will prompt the user to finish allocating the payment methods. In addition, if the customer has not yet been verified at this point, name capture window **90** will pop open to capture the customer's billing-related information.

FIG. **15** displays a preferred object inheritance structure for some preferred objects associated with billing module **36**. These objects include a payment type object **122**, a coupon object **124**, a recovery coupon object **126**, a credit card object **128**, a Visa® credit card object **130**, Mastercard® credit card object **132**, and an American Express® credit card object **134**. Each object defines a set of related functions for operating on payment method-related information. The preferred functions for the objects associated with billing module **36** include adding a new payment type, fetching the history of an existing payment type, deleting a selected payment method, deleting an order, retrieving the allocated dollar amounts and percentages based on the allocations selected by the customer, and updating those allocations.

All of the functions defined by the object, however, are not necessarily needed for each object type. For example, no history information exists for payment with a coupon. The "fetch history" and "delete history" functions of coupon object **124** are therefore, preferably inactive, as is illustrated by the non-highlighted nature of these functions on coupon object **124**. All of the other functions of coupon object **124** are applicable and are thus, preferably highlighted.

FIG. **16** represents a detailed window data flow for billing module **36**. In particular, FIG. **16** shows the interaction between an allocate window **120** and credit card object **128**, coupon object **124**, and Recovery coupon **126** of FIG. **15**. For example, if the user is done allocating the payment methods for the order, upon selecting the done button **119** from window **120**, system **10** is directed to the update allocate function of the appropriate objects.

ORDER CAPTURE MODULE

Order Capture module **34** identifies the way in which a user may capture an order for an offer from a customer. A detailed dialog flow for the order capture module **34** is shown in FIG. **17**. In order to capture an order, the user must be able to pull up the offer from the system. In this regard, order capture module **34** is made up of four functions, namely source searching, page searching, offer lookup, and manufacturer lookup. These functions may be represented on toolbar **52** of the user interface of FIG. **7** by source search button **54***e*, page search button **54***f*, order lookup button **54***g*, and manufacturer lookup button **54***h*. Each of these functions will now be described in detail below.

Upon activation of source search button **54***e*, a source search window **140** as shown in FIG. **18** pops open and permits the user to select the source in which the offer being ordered is located. Source search window **140** defines source search criteria data capture fields for capturing a search criteria from the user. As is shown from source search window **140** of FIG. **18**, three search criteria data capture fields are preferably identified. These include a source type search criteria data capture field **142**, a source date search criteria data capture field **148** and a source group search criteria data capture field **150**.

Source type search criteria data capture field **142** identifies all of the source types used by a particular business to expose their product line. Such source types may include catalogs, magazines, billboards, infomercials and the like. It is understood that the search criteria will vary depending upon the source type selected. Upon selecting a source type, window **140** displays at least one visual image **146** depicting a representation of a source associated with the source type currently selected. For example, in the case of catalogs, image **146** preferably represents a cover page. In a preferred form, a plurality of visual images **146** as shown in FIG. **18** are provided. The number of visual images **146** displayed, however, is not critical.

Source search window **140** also identifies date search criteria data capture field **148** and source group search criteria data capture field **150**. In the preferred system, after a source type has been selected, the user can further filter or expand the list of sources by changing the source date captured in source date search criteria data capture field **148** and/or selecting a source group from source group search criteria data capture field **150**.

Buttons **147** are preferably provided for allowing the user to scroll through the sources one at a time, preferably in an ascending or descending date order, or to immediately display the first or last source within the search criteria.

If the user wishes to search the selected source by page, a page search window **190** as shown in FIG. **19** may be opened so that the user can continue to process the order. Page search window **190** preferably displays a visual image **192** representing the selected source, as well as a plurality of visual images **194** representing pages from the source. Upon opening window **190**, two visual images **194** representing the first two pages from the selected source are preferably displayed. The number of visual images **194** displayed, however, is not critical.

Page search window **190** also preferably allows the user to automatically change to a different source without having to return to source search window **140**. This is accomplished by selecting a source from source data capture field **193**. The user may also automatically return to source search window **140** by selecting the source button **195**.

Buttons **196** may be provided on page search window **190** for allowing the user to scroll through the pages of the selected source either one at a time in an ascending or descending page number order, or to immediately display the first or last page. The user also preferably has the option to automatically pull up a particular page within the source while still staying in page search window **190** via a page data capture field **198**. If the desired page is found, the user may view the offers associated with that page by entering the corresponding page number in a go to page data capture field **200**, which automatically opens a page offer window **210** as is shown in FIG. **20**.

Page offer window **210** preferably displays a visual image **212** of the particular page selected from page search window **190**, and identifies all of the offers associated with that page in offer data capture field **211**. A page number data capture field **209** and corresponding go to button **208** may be provided to allow the user to selectively change pages.

Information about each offer identified in offer data capture field **211** may be displayed in the page description data capture field **213**. Additional information about the offers, such as warranty information, may be automatically displayed by selecting the notes button **215**. Offers can be ordered directly from this window by selecting the add button **214**.

An offer is an item presented to the customer at a specific price. An offer is represented by an offer object **141** as is shown in FIG. **21**. As is shown in FIG. **21**, the offers available through the preferred system include a single-SKU offer object **143**, a multi-attribute offer object **144**, and a multi-SKU offer object **145**. Single SKU offer object **143** refers to one offer having one SKU. Multi-attribute offer object **144** refers to one offer available with various attributes, such as color or size. Multi-SKU offer object **145** refers to a combination of two offers, such as a shirt and tie combination.

However, in order to interface with back-end systems **8**, each of the offers ordered by the customer must correspond to a single SKU. Accordingly, the preferred system translates each combination of attributes for the offer and each combination of two offers into a unique SKU. With such a configuration, new offer objects may be added without affecting existing offer objects. Rather, the new offer object absorbs the properties of the existing offer objects. The only change that is necessary is a change to the inventory business rules.

Upon selecting add button **214** from page offer window **210**, if a multi-attribute offer is selected by the user, a multi-attribute offer window **180** as shown in FIG. **22** pops open. A preferred multi attribute offer window **180** displays all offer combinations associated with a particular offer. An offer description is preferably displayed in offer description data capture field **181**. Each of the offer's attributes are preferably displayed in attribute data capture fields **186**. Each attribute is preferably associated with corresponding attribute button **184**. For the example shown in FIG. **22**, four attributes are identified, namely color, fit, neck and sleeve. The number and type of attributes displayed changes with the offer. In a preferred form, the system allows up to fifteen (15) attributes. Such attributes may include color, size, fit, fabric, collar type, length, width, neck size, sleeve length and pattern.

In the preferred system, the user may obtain more information on a particular attribute from additional windows by selecting the particular attribute button. For example, upon selection of the fit attribute button, a fit attribute window **300** as shown in FIG. **23** pops open and displays a picture and/or text to further describe the selected fit.

Referring back to FIG. **22**, an offer combination data capture field **187** displays the offer associated with all of the offer combinations available for the selected offers. This field is continually filtered as the user selects the attributes requested by the customer. After one offer is selected from offer combination data capture field **187**, the user can add the offer to the order by selecting the add button **189**. A visual image of the offer may preferably be displayed by selecting the offer image button **182**. The visual image depicted can represent the offer having the selected attribute choices. This helps the user describe the offer when the customer requests information about a particular offer.

Referring back to FIG. **19**, page search window **190** may also provide a source offer button **191** which, when selected, opens a source offer window **160** as shown in FIG. **24**. Source offer window **160** may also be accessed by selecting a source from source search window **140** of FIG. **18**. Source offer window **160** typically is used to display all of the offers available through the selected source, along with their respective prices at a data capture field **163**. Additional information about each offer identified in data capture field **163** may be displayed in the source description data capture field **167**. A visual image **161** of the source selected is also preferably displayed. Window **160** also preferably allows the user to change to a different source without having to return to source search window **140** by selecting a source from source data capture field **165**. The offers associated with a source are automatically updated as the source changes. Selection of the source button **168** automatically takes the user back to source search window **140**.

Upon selecting one of the offers identified in source offer window **160**, the user preferably has two choices. First, the user may obtain information about the selected offer by selecting the offer information button **166**. Second, the user may order the selected offer by selecting the add button **169**. As described earlier herein, if a multi-attribute offer is selected by the user, a multi-attribute offer window **180** as shown in FIG. **21** pops open.

As previously mentioned, the user can also search for offers by conducting an offer search. Upon activation of

offer lookup button **52***g*, an offer search window **220** as is shown in FIG. **25** opens. Offer search window **220** is typically used to search for offers based on customer criteria and is often used when a customer does not have a specific source from which to refer. Window **220** preferably identifies offer search criteria data capture fields, which preferably include offer category search criteria data capture field **222**, offer class search criteria data capture field **224** and offer sub-class search criteria data capture field **226**.

Each offer class search criteria identified in offer class search criteria data capture field **224** is normally associated with at least one offer category search criteria. Preferably, each offer class search criteria identified in offer class search criteria data capture field **224** is associated with a plurality of offer category search criteria so as to define a separate search path for each offer category/offer class association. With such a configuration, an offer may be searched in a multitude of ways. Offer sub-class preferably has the same relationship to offer class.

Upon entering window **220**, normally only offer category search criteria data capture field **222** is filled with information. The user preferably has two options; namely to capture a keyword in the keyword data capture field **228** or to select an offer category search criteria from offer category search criteria data capture field **222**. When the user selects an offer category search criteria and selects the search button **221**, all of the offer class search criteria that are associated with that offer category search criteria are preferably automatically displayed in offer class search criteria data capture field **224**. After an offer class search criteria is selected and search button **221** is selected, all of the offer sub-class search criteria associated with that offer class search criteria are automatically displayed in offer sub-class search criteria data capture field **226**.

When the user enters a keyword into keyword data capture field **228** and selects the keyword button **230**, the associated offer categories, offer classes and offer sub-classes are preferably automatically displayed on window **220**. If there is more than one offer category, the offer class search criteria data capture field **224** and offer sub-class search criteria data capture field **226** will typically be empty until the user selects an offer category search criteria. If there is more than one offer class, offer sub-class search criteria data capture field **226** will typically be empty until the user selects an offer class search criteria. If an offer sub-class exists for the offer class search criteria selected, it normally must be selected before searching for associated offers.

Offers uncovered through the offer search are automatically displayed in the offer data capture field **232**. The user can further filter the list of offers by entering information in the key features data capture field **234** or the manufacturer data capture field **236** and selecting the filter button **235**. Information about a particular offer can be displayed by selecting the offer information button **238**. When a specific offer is selected, the add button **240** may be used to add the offer to the order. Upon selecting add button **240**, an order quantity window (not shown) preferably opens so that a quantity can be captured. In this manner, the user is able to confirm the quantity of an offer being purchased by the customer before actually adding it to the order. Window **220** closes upon selecting the done button **239**.

As previously mentioned, the user may also search for offers by performing a manufacturer search. Upon selection of manufacturer search button **52h**, the manufacturer search window **250** as shown in FIG. **26** automatically opens. Through window **250**, the user can search for an offer based on the offer'manufacturer. This window is typically used when a customer does not have a specific source from which to refer.

When manufacturer search window **250** is opened, a list of manufacturer search criteria are identified in the manufacturer search criteria data capture field **252**. The user normally must select a manufacturer search criteria before searching for associated offers. If the user chooses to search, all offers made by the selected manufacturer are displayed in the offer data capture field **254**. The user preferably can further filter the list of offers displayed in offer data capture field **254** by entering searching criteria in the key feature data capture field **256**.

Once an offer has been selected, further information about that offer can preferably be obtained by selecting the notes button **255**. The user may also add the selected offer by selecting the add button **257**. Upon selecting add button **257**, an order quantity window (not shown) automatically opens so that a quantity can be captured. In this manner, the user is able to confirm the quantity of an offer being purchased by the customer before adding it to the order.

Referring back to FIG. **17**, order capture module **34** also preferably provides several other advantageous features. It allows the user to keep track of lost demand via a lost demand reason codes window (not shown). This window displays a list of possible reason codes the user can select to account for a decrease in demand, such as credit card decline or customer request. The list of reason codes displayed is governed by the action that precipitated the lost demand (e.g. canceling of an order).

Order capture module **34** also preferably provides for the cross-sell of additional product offering which may be suggested to the customer based on the customer's current order. Order capture module **34** may also be adapted to base such cross-selling on the customer's past purchasing history and general demographic information. When the customer places an order, an add-on window **400** may be opened, and displays a description of any add-on products that are associated with the offer just ordered by the customer. For example, if a customer ordered a television, add-on window **400** may display a video cassette recorder (VCR) and a VCR/TV adapter kit.

If an ordered product is not available, a backorder window **402** preferably opens and provides the customer with four choices. He or she may order the quantity that is available and accept the backorder quantity to be shipped at a later time, order only the available quantity, choose to order an alternate or substitute offer (if one exists), or cancel the whole quantity of the original product. If the customer chooses to order a substitute, substitution window opens and displays a list of available substitutes. When an item is out of stock, either the item will automatically be substituted with the same item from a different vendor (for the same price), or a list of substitute items will be displayed. The customer may choose to substitute the original item with an item displayed in the list or choose from the list of alternate items.

SHIPPING MODULE

Shipping module **38** identifies the way in which offers ~~ordered by a customer may be~~ shipped. A detailed dialog flow for shipping module **38** is shown in FIG. **27**. Upon selecting shipping button **54**d from toolbar **52** of the user interface shown in FIG. **7**, a shipping window **270** as is shown in FIG. **28** opens. Shipping window **270** captures ship to addresses in ship to data capture field **272** and assigns ordered offers to those addresses. The user can elect to enter the customer's address information directly (i.e., in the case of a different ship to address than the bill to address) or he or she can use one of the previous ship to addresses, if one exists. The user can also enter one or more new ship to addresses. The entire quantity of an ordered offer can be assigned to a single address or partial quantities can be assigned to multiple addresses.

Shipping window **270** also displays shipping information in data capture fields **276**. Such information includes the shipping carrier and shipping service level, the shipping and arrival date and the cost for shipping the order with that carrier and at that service level. As previously stated herein, the preferred system automatically determines the best shipping method based in part on the offer being ordered, the shipping address, and the shipping carrier and shipping service level.

From shipping window **270**, the user can select the shipping calendar button **278** to open a shipping calendar window **280**, as is shown in FIG. **29**. Shipping calendar window **280** preferably displays each shipment for the current address, one at a time at data capture field **282**. A shipment is defined as all ordered offers that are being shipped on the same date, to the same address, using the same shipping carrier and shipping service level. The order date, shipping date and delivery date are either alphanumerically or graphically identified on shipping calendar window **280** so that they may be easily identified by the user. Holidays and non-delivery days for the selected carrier and service level are also identified. For example, for the shipment displayed on shipping calendar window **280** of FIG. **29**, the order date is Jun. 30, 1994, the ship date is Jul, 1, 1994 and the arrival date is Jul. 14, 1994. Non-delivery dates include all Sundays.

The carrier and service level for the current shipment can be assigned by clicking on the ship via button **284** on shipping window **280** of FIG. **29**. The user can choose from all carrier and service level combinations that have passed all validation for the offers in the current shipment and for the ship to address. External shipping instructions that are needed in order for the shipment to arrive to the customer may also be provided to the user.

Shipping window **270** of FIG. **28** also contains a list of all of the ordered offers that have not yet been assigned to a shipment at data capture field **274**. The user can assign one or more of the items identified in data capture field **274** to a ship to address by clicking on the assign selected button **286**. If all of the ordered offers are to be shipped to the same address, the user may assign the items that address by clicking on the select all button **288**. In either case, the user is taken to a ship to contents window **290** as shown in FIG. **27**, which displays the selected ship to address and the offers assigned to each shipment for this address. If there are multiple shipments for this address, all shipments will be displayed. The user can delete any offer displayed. Offers can also be transferred from this address to another address. In the case of multiple shipments, a quantity to assign window **292** displays the quantity available to assign or transfer. The user is allowed to modify the quantity that should be assigned as long as it is less than or equal to the quantity available.

Order entry system **10** also provides other advantageous functions and features in addition to those identified in the discussion of the modules of FIG. **7**. One such function is a marketing function, the concept of which is illustrated in FIG. **30**. FIG. **30** shows a user interface represented by a window 500. Window 500 preferably defines a plurality of user-initiated events and a plurality of application points corresponding to the user-initiated events. Such user-initiated events may include selecting a button 501, entering a window 502, exiting a window 503 and capturing data in a data capture field 504.

An action may be assigned to any of the application points corresponding to such user-initiated events. Marketing objects are provided by the preferred order entry system for dynamic invocation of the action upon user initiation of the event corresponding to the application point assigned to the action. With such a configuration, the user is provided with the action at a point during the placement of the order at which the action is needed.

FIG. 31 shows a preferred application point maintenance window 520. Application point maintenance window 520 is designed to allow a system administrator to maintain existing application point descriptions and attributes, as well as to create a new application point record when one is required for use in the system application. The system preferably automatically assigns the next sequential application point. The application point number assignment process also allows the system administrator to distribute an application point that has not been used previously thereby preventing application point.

Referring to FIG. 31, application point maintenance window 520 includes a select button 522 which, when selected, causes highlighted application point information to be displayed in the maintenance area of window 520 to allow the user to edit information or use the information as a basis for a new application point. Selection of the new button 524 causes the information displayed in the maintenance area to be saved to the database as a new application point. Selection of the update button 526 updates the information displayed in the maintenance area that has been modified by the user. The clear button 528 clears all of the information in the maintenance area. Any changes that have been made to the information there that have not been updated to the database using new button 524 or update button 526 are not saved.

System 10 preferably allows the user to selectively activate and de-activate any of the application points associated with the marketing object, as well as to selectively change the assignment of an action. In addition, the preferred system allows the user to assign the application points such that a plurality of user experience levels is achieved, as well as to set the frequency at which the action is initiated.

The action that is initiated may be any type of event. One such action can be to display a message. The message can be instructions to the user to help him or her in the placement of an order. In this regard, marketing objects preferably have the capability to display the message in a plurality of languages.

The action may also be providing a marketing promotion. Such promotions may include product-to-product cross sell, free gifts, dollar or percentage off a line item, the order, the shipping or the handling coupons, a gift, a coupon, a discount, a payment type discount, a shipment service level discount, and a shipping and handling discount. They are typically driven by order and customer characteristics. Order characteristics may include total dollar amount, total line item, line quantity, and payment method to name a few. Customer characteristics may include promotion success rate, average order size, length of customer, order frequency, and payment history to name a few.

FIG. 35 illustrates a promotion maintenance window 580. Promotion maintenance window 580 displays existing marketing promotions in promotion display data capture field 586. The user preferably may look up promotions by any of four options identified in the options data capture field 588, namely, company, source, offer number or club. The information retrieved from a look up function may be limited by selecting a specific company or promotion effort. Once a promotion is selected, the user may assign the promotion to a particular application point via the assign to application point button 582 or choose to edit the promotion via the edit button 584.

FIG. 36 illustrates a promotion process maintenance window 600. Promotion process maintenance window 600 provides the user with the ability to assign promotions to specific application points. An application point determines where in the application a particular promotion will be prompted. Preferably, the user is also able to specify whether the promotion is active or inactive via active indicator data capture field 602, and the order in which the promotions will be processed via the process order data capture field 604.

FIG. 32 shows a marketing promotions window 530 which allows the user to select current marketing promotions based on the type of promotion structure selected. Once the proper promotion structure's components have been selected, the user may look up existing promotion action types assigned to the promotion by selecting lookup button 532, or assign new promotion action types to the promotion structure components by selecting assignment button 534.

FIG. 33 shows a marketing objects window 540 which allows the user to add, modify and delete objects as well as assign marketing objects to application points. When opened, window 540 captures all companies currently set up on the system's company data table and all marketing objects for the first company on the list. By selecting a new company, the marketing objects in marketing objects window 540 will be updated to reflect only those marketing objects for the company selected. From here, the user can delete a marketing object via the delete button 542, update a marketing object via the edit button 544, or assign a marketing object to an application point. The new button 548 can be pressed to set up new marketing objects.

From marketing objects window 540, the user may open a marketing object maintenance window 550 as shown in FIG. 34. Window 540 allows the user to enter the marketing object's file name and description, and to add, edit and delete marketing object/application point cross reference records. These records are used to execute marketing objects when certain application points are triggered in the system's application. When first opened, window 550 displays the object information that was selected from marketing objects window 540, current application points that are tied to the selected marketing object, and available application points that may be tied to the selected marketing object. The available application points are a subset of all application points available. The application points are filtered by selecting a function/window combination from function data capture field 556 and window data capture field 558.

Once opened, the user selects either an existing cross reference record from the current application points identified in the current application point data capture field 560, or a new application point. All marketing objects that are tied to the application point selected are automatically displayed in the current object cross references area data capture field 562. This will ensure that the user has visibility to all objects that will be processed when the application point is triggered. To delete cross references, the user selects a row on the current object cross references area and presses the delete button **564**. This will delete the row off the screen. Once all the updates to cross references have been made, including deletions, the user presses the update button **566** to save the changes to the database. Pressing the cancel button **554** clears all changes made without saving.

We claim:

1. A computerized system for the placement of an order by a user for at least one of a plurality of offers via a terminal having a display, the system comprising:

(a) storage means for storing offer information relating to the plurality of offers, and for storing electronic reproductions of a plurality of offer sources, each of the plurality of offer sources containing at least one of the plurality of offers;

(b) source searching means for locating one or more of the electronic reproductions of the plurality of offer sources, the source searching means comprising:

(i) source display means for displaying a plurality of electronic offer source images, wherein the plurality of electronic offer source images represent identifying portions of the plurality of offer sources;

(ii) source filtering means for receiving source search criteria, and for causing the source display means to display those of the plurality of electronic offer source images which meet the source search criteria; and

(iii) source selection means for allowing selection of one of the plurality of electronic offer source images which represents an identifying portion of one of the plurality of offer sources;

(c) offer searching means for locating the at least one of the plurality of offers associated with the selected electronic offer source image, the offer searching means comprising:

(i) source segment display means for displaying, in response to offer search criteria, one or more electronic images of a portion of the offer source represented by the selected electronic offer source image; and

(ii) offer selection means for allowing selection of the at least one of the plurality of offers associated with the one or more electronic images displayed,

whereby the user is enabled to locate the at least one of the plurality of offers by searching through an electronic equivalent of the offer source used by the person placing the order.

2. The computerized system of claim **1**, wherein the source filtering means comprises source type filtering means for receiving source type search criteria, and for causing the source display means to display those of the plurality of electronic offer source images which meet the source type search criteria.

3. The computerized system of claim **2**, wherein the source type filtering means comprises:

source type display means for displaying a selectable list of source types; and

source type selection means for allowing selection of one of the source types in the selectable list.

4. The computerized system of claim **3**, wherein the source types in the selectable list are selected from a group comprising one or more of catalogs, space ads, billboards, infomercials and television.

5. The computerized system of claim **2**, wherein the source filtering means further comprises source group filtering means for receiving source group search criteria, and for causing the source display means to display those of the plurality of electronic offer source images which meet the source type search criteria and the source group search criteria.

6. The computerized system of claim **5**, wherein the source group filtering means comprises:

source group display means for displaying a selectable list of source groups; and

source group selection means for allowing selection of one of the source groups in the selectable list.

7. The computerized system of claim **1**, wherein the source filtering means comprises date filtering means for receiving date search criteria, and for causing the source display means to display those of the plurality of electronic offer source images which meet the date search criteria.

8. The computerized system of claim **7**, wherein the date filtering means comprises date range means for receiving date search criteria corresponding to a range of source issue dates.

9. The computerized system of claim **7**, wherein the date filtering means comprises date order means for receiving date search criteria corresponding to an ascending date order and a descending date order.

10. The computerized system of claim **1**, wherein the source filtering means comprises means for causing the source display means to display less than all of the plurality of electronic offer source images which meet the source search criteria at a time.

11. The computerized system of claim **10**, wherein:

the plurality of electronic offer source images which meet the source search criteria include a first offer source image and a last offer source image; and

the source filtering means comprises means for automatically locating at least one of the first offer source image and the last offer source image.

12. The computerized system of claim **10**, wherein the source searching means further comprises scroll display means for scrolling through and displaying the remaining ones of the plurality of electronic offer source images which meet the source search criteria.

13. The computerized system of claim **1**, wherein the offer selection means comprises:

offer choice display means for displaying a list of the plurality of offers associated with the one or more electronic images displayed by the source segment display means; and

offer choice selection means for allowing selection of at least one of the plurality of offers from the list.

14. The computerized system of claim **1**, wherein:

the portion of the offer source corresponds to one or more pages of a multi-page offer source; and

the source segment display means comprises page view means for displaying the electronic images of the one or more pages.

15. The computerized system of claim **14**, wherein the source segment display means further comprises page scrolling means for scrolling through all of the pages of the offer source represented by the selected electronic offer source image.

16. The computerized system of claim **14**, wherein the source segment display means further comprises page indication means for specifying a particular one of the pages to be displayed.

17. The computerized system of claim **1**, wherein the source segment display means further comprises source segment scrolling means for scrolling through all of the

portions of the offer source represented by the selected electronic offer source image.

18. A computerized method for the placement of an order by a user for at least one of a plurality of offers via a terminal having a display, the method comprising the computer-executed steps of:

(a) storing offer information relating to the plurality of offers, and storing electronic reproductions of a plurality of offer sources, each of the plurality of offer sources containing at least one of the plurality of offers;

(b) locating one or more of the electronic reproductions of the plurality of offer sources, the step of locating one or more of the electronic reproductions comprising the steps of:

(i) displaying a plurality of electronic offer source images, wherein the plurality of electronic offer source images represent identifying portions of the plurality of offer sources;

(ii) entering source search criteria;

(iii) displaying those of the plurality of electronic offer source images which meet the source search criteria; and

(iv) selecting one of the plurality of electronic offer source images which represents an identifying portion of one of the plurality of offer sources;

(c) displaying one or more electronic images of a portion of the offer source represented by the selected electronic offer source image, in response to offer search criteria; and

(d) selecting the at least one of the plurality of offers which is associated with the one or more electronic images displayed, whereby the user is enabled to locate the at least one of the plurality of offers by searching through an electronic equivalent of the offer source used by the person placing the order.

19. The computerized method of claim 18, wherein:

the step of entering source search criteria comprises the step of entering source type search criteria; and

the step of displaying those of the plurality of electronic offer source images which meet the source search criteria comprises the step of displaying those of the plurality of electronic offer source images which meet the source type search criteria.

20. The computerized method of claim 19, wherein the step of entering source type search criteria comprises the steps of:

displaying a selectable list of source types; and

selecting one of the source types in the selectable list of source types.

21. The computerized method of claim 20, wherein the source types in the selectable list are selected from a group comprising one or more of catalogs, space ads, billboards, infomercials and television.

22. The computerized method of claim 19, wherein:

the step of entering source search criteria further comprises the step of entering source group search criteria; and

the step of displaying those of the plurality of electronic offer source images which meet the source search criteria comprises the step of displaying those of the plurality of electronic offer source images which meet the source type search criteria and the source group search criteria.

23. The computerized method of claim 22, wherein the step of entering source group search criteria comprises the steps of:

displaying a selectable list of source groups; and

selecting one of the source groups in the selectable list of source groups.

24. The computerized method of claim 18, wherein:

the step of entering source search criteria comprises the step of entering date search criteria; and

the step of displaying those of the plurality of electronic offer source images which meet the source search criteria comprises the step of displaying those of the plurality of electronic offer source images which meet the date search criteria.

25. The computerized method of claim 24, wherein the step of entering date search criteria comprises the step of entering a range of dates corresponding to a range of source issue dates.

26. The computerized method of claim 24, wherein the step of entering date search criteria comprises the step of entering date order search criteria for searching by ascending dates and descending dates.

27. The computerized method of claim 18, wherein the step of displaying those of the plurality of electronic offer source images which meet the source search criteria comprises the step of displaying less than all of the electronic offer source images which meet the source search criteria at one time.

28. The computerized method of claim 27, wherein the step of displaying less than all of the electronic offer source images further comprises the step of scrolling through and displaying the remaining ones of the electronic offer source images which meet the source search criteria.

29. The computerized method of claim 18, wherein the step selecting the at least one of the plurality of offers comprises the steps of:

displaying an offer list of the plurality of offers associated with the one or more electronic images displayed by the step of displaying one or more electronic images; and

selecting at least one of the plurality of offers from the offer list.

30. The computerized method of claim 18, wherein:

the portion of the offer source corresponds to one or more pages of a multi-page offer source; and

the step of displaying one or more electronic images of a portion of the offer source comprises the step of displaying electronic page images of the one or more pages of the multi-page offer source.

31. The computerized method of claim 30, wherein the step of displaying electronic page images further comprises the step of scrolling through all of the pages of the multi-page offer source.

32. The computerized method of claim 31, wherein the step of scrolling through all of the pages includes the step of scrolling through the pages one page at a time in an ascending order.

33. The computerized method of claim 31, wherein the step of scrolling through all of the pages includes the step of scrolling through the pages one page at a time in a descending order.

34. The computerized method of claim 31, wherein the step of displaying electronic page images further comprises the step of specifying a particular one of the pages to be displayed.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : **5,832,459**

DATED : **NOVEMBER 3, 1998**

INVENTOR(S) : **CAMERON ET AL.**

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Col. 5, line 4: "direction" should read --directing--

Col. 5, line 6: insert --devices-- after the word "entry"

Col. 6, line 1: "organization' " should read --organization's--

Col. 6, line 58: "**SB**" should read --**5B**--

Col. 9, line 4: "customer' " should read --customer's--

Col. 10, line 15: "customer' " should read --customer's--

Col. 10, line 56: "customer' " should read --customer's--

Col. 11, line 27: "customer' " should read --customer's--

Col. 17, line 1: "offer' " should read --offer's--

Signed and Sealed this

Tenth Day of April, 2001

Attest:

Nicholas P. Godici

NICHOLAS P. GODICI

Attesting Officer

Acting Director of the United States Patent and Trademark Office

Google™ Advanced Search Preferences Language Tools Search Tips

stamps.com inc [Google Search]

Web · Images · Groups · Directory · News ·

Searched the web for **stamps.com inc**. Results **1** - **10** of about **27,100**. Search took **0.36** seconds.

Stamps.**com**: Print Internet Postage, NetStamps and Shipping Labels ...
... In cooperation with Visit Microsoft Office Services, Copyright ©
1998-2003 **Stamps**.**com Inc**. All Rights Reserved. Microsoft and the ...
Description: Develops internet-based postal service delivery system. (Nasdaq: STMP)..
Category: Business > Major Companies > Publicly Traded > S
www.stamps.com/ - 16k - Cached - Similar pages - Stock quotes: STMP

STMP: Profile for **STAMPS.COM INC** - Yahoo! Finance
... Quotes & Info, Enter Symbol(s): eg YHOO, ^DJI Symbol Lookup | Finance Search.
Stamps.**com Inc** (STMP), At 4:00PM ET: 5.42 0.09 (1.69%) Reuters. ... **Stamps**.**com Inc**. ...
biz.yahoo.com/p/s/stmp.html - 24k - Cached - Similar pages

STMP: Summary for **STAMPS.COM INC** - Yahoo! Finance
... Enter Symbol(s): eg YHOO, ^DJI Symbol Lookup | Finance Search. **Stamps**.**com Inc**
(STMP), On Nov 21: 5.79 0.10 (1.76%) Reuters. ... , **Stamps**.**com Inc**. Financials. ...
finance.yahoo.com/q?s=stmp&d=t - 28k - Cached - Similar pages
[More results from yahoo.com]

FindLaw Business Contracts : By Alpha : S : **Stamps**.**com Inc**.
... Contracts Home : By Alpha : S : **Stamps**.**com Inc**. **Stamps**.**com Inc**. ... (Dec 16, 1998);
Agreement and Plan of Merger - **Stamps**.**com Inc**. and iShip.com **Inc**. ...
contracts.corporate.findlaw.com/alpha/s/1748.html - 37k - Cached - Similar pages

FindLaw - Distributor Agreement - **Stamps**.**com Inc**. and Office ...
... This Distributor Agreement (the "Agreement") is made as of this 15th day of January,
1999 (the "Effective Date"), by and between **Stamps**.**com Inc**., a Delaware ...
contracts.corporate.findlaw.com/agreements/ stamps/office.distrib.1999.01.15.html - 100k - Cached - Similar pages
[More results from contracts.corporate.findlaw.com]

FindLaw Business Contracts : By Alpha : S : **Stamps**.**com Inc**.
... Tech Deals, Powered by FindLaw. Contracts Home : By Alpha : S : **Stamps**.**com Inc**.
Stamps.**com Inc**. ... (Dec 16, 1998); Agreement and Plan of Merger - **Stamps**.**com Inc**. ...
techdeals.startup.findlaw.com/alpha/s/1748.html - 30k - Cached - Similar pages

Stamps.**com Inc**. - Fact Sheet - Hoover's Online
Stamps.**com Inc**. (NASDAQ: STMP). 3420 Ocean Park Blvd., Ste. 1040 ... Partners.
Buy Reports and Books. D&B Business/Credit Reports: **Stamps**.**com Inc**. ...
www.hoovers.com/stamps.com/--ID__58250--/ free-co-factsheet.xhtml - 33k - Cached - Similar pages

Stamps.**com Inc**. - SEC Filings - Hoover's Online
Stamps.**com Inc**. (NASDAQ: STMP). ... STMP, **STAMPS COM INC**, 10-Q, Quarterly report which
provides a continuing view of a company's financial position, 11/13/2003, Filing. ...
www.hoovers.com/stamps.com/--ID__58250,Page__1,Sort__D--/ free-co-sec.xhtml - 39k - Cached - Similar pages
[More results from www.hoovers.com]

Stamps.**com Inc**.
Stamps.**com Inc**. (NASDAQ-NM:STMP), Quote: Nov-21 6:03 PM ET. Industry Sector:
Catalog & Mail Order Company Ranking within Sector: 8. Last Trade, 5.79. ...
www.retailnet.com/company.cfm?ID=stmp - 16k - Cached - Similar pages

[PDF] **Stamps**.**com Inc**.
File Format: PDF/Adobe Acrobat

Google™ Advanced Search Preferences Language Tools Search Tips

internet postage [Google Search]

Web · Images · Groups · Directory · News ·

Searched the web for **internet postage**. Results **11** - **20** of about **1,110,000**. Search took **0.10** seconds.

internet postage
Offers **internet postage** product or service and **internet** stamps and also pc **postage**. **internet postage**. **internet postage**. Offers ...
www.postage--stamp.com/internet-postage.html - 6k - Cached - Similar pages

Stamps.com -- **Internet Postage** Stamps Online
... Pack it, weigh it, ship it. Stamps.com® **Internet Postage**. ... With our **Internet Postage** service you can buy and print your own **postage**, right from your PC. ...
www.gochicagoland.com/estamps.htm - 27k - Cached - Similar pages

Internet Postage Providers Add 'Stealth' Feature for Online ...
... 1528-6703. **Internet Postage** Providers Add 'Stealth' Feature for Online Sellers By Ina Steiner AuctionBytes.com May 18, 2003. Many ...
tinyurl.com/eygj - 20k - Cached - Similar pages

A Discount for **Internet Postage**?
Postal Rate Commission mulling over discounted rates for **Internet postage**; Stamps.com pleads for rate's creation. The World's Largest Wi-Fi Business Event ...
www.internetnews.com/ec-news/article.php/511191 - 66k - Cached - Similar pages

> Stamps.com Sees **Internet Postage** As 'Viable'
> **Internet postage** firm is quick to congratulate itself for not going down the road taken by competitor E-Stamp. The World's Largest Wi-Fi Business Event ...
> www.internetnews.com/ec-news/article.php/520461 - 59k - Cached - Similar pages
> [More results from www.internetnews.com]

E-Commerce News: Pitney Bowes Joins **Internet Postage** Crowd
... reprints.shtml. E-BUSINESS SECTION. Pitney Bowes Joins **Internet Postage** Crowd. By Rob Conlin E-Commerce Times March 29, 2000. Mail ...
www.ecommercetimes.com/perl/story/2851.html - 37k - Cached - Similar pages

Publications: **Internet Postage** - Becker & Poliakoff
Internet Postage by Allison Kimberly Hift, Becker & Poliakoff, PA.
Did you ever think the day would actually arrive when you could ...
www.becker-poliakoff.com/publications/ article_archive/internet_postage.htm - 22k - Cached - Similar pages

How to eDo It - Learn How **Internet Postage** Works
Internet Postage Information. ... If you have a computer, printer, and **Internet** connection you can print **postage**, from your desktop - 24 hours a day. ...
www.howtoedoit.com/services/pcpostage.htm - 11k - Cached - Similar pages

e-stamp The **Internet Postage** Company
... Home > Communications, Dot Coms, Computers > Dot Coms > e-stamp The **Internet Postage** Company. e-stamp The **Internet Postage** Company. ...
www.scripophily.net/esinposcom.html - 35k - Cached - Similar pages

INTERNET ARCHIVE
WaybackMachine

Enter Web Address: http:// All Take Me Back Adv. Search Compare Archive Pages

Searched for **http://www.stamps.com** **248** Results

Note some duplicates are not shown. See all.
* denotes when site was updated.

Search Results for Jan 01, 1996 - Dec 04, 2003

1996
0 pages

1997
0 pages

1998
3 pages

- Nov 11, 1998 *
- Dec 01, 1998 *
- Dec 12, 1998 *

1999
5 pages

- Jan 25, 1999 *
- Apr 28, 1999 *
- Oct 01, 1999 *
- Oct 12, 1999
- Nov 17, 1999

2000
28 pages

- Feb 29, 2000 *
- Mar 01, 2000
- Mar 03, 2000
- Mar 03, 2000 *
- Mar 04, 2000
- Apr 07, 2000 *
- May 10, 2000 *
- May 10, 2000 *
- May 11, 2000 *
- May 12, 2000
- May 20, 2000 *
- May 20, 2000 *
- Jun 13, 2000 *
- Jun 19, 2000 *
- Jun 20, 2000 *
- Jun 21, 2000
- Jun 22, 2000
- Jul 06, 2000 *
- Aug 15, 2000 *
- Oct 06, 2000 *
- Oct 08, 2000 *
- Oct 08, 2000 *
- Oct 17, 2000 *
- Oct 18, 2000
- Oct 19, 2000
- Nov 09, 2000 *
- Nov 19, 2000 *
- Dec 02, 2000 *

2001
143 pages

- Jan 18, 2001 *
- Jan 19, 2001
- Jan 27, 2001
- Feb 06, 2001
- Feb 26, 2001 *
- Mar 01, 2001 *
- Mar 02, 2001
- Mar 09, 2001
- Mar 30, 2001
- Apr 05, 2001 *
- Apr 29, 2001 *
- May 05, 2001 *
- May 07, 2001
- May 12, 2001 *
- May 16, 2001
- May 17, 2001
- May 18, 2001
- May 19, 2001
- May 22, 2001
- May 31, 2001 *
- Jun 01, 2001
- Jun 02, 2001
- Jun 03, 2001
- Jun 06, 2001 *
- Jun 07, 2001
- Jun 11, 2001
- Jun 14, 2001
- Jun 26, 2001
- Jun 27, 2001
- Jun 29, 2001
- Jun 30, 2001
- Jul 01, 2001
- Jul 08, 2001
- Jul 10, 2001
- Jul 13, 2001 *
- Aug 25, 2001 *
- Oct 09, 2001 *
- Oct 09, 2001 *
- Oct 09, 2001 *
- Oct 10, 2001 *
- Oct 10, 2001 *
- Oct 11, 2001 *
- Oct 11, 2001 *
- Oct 12, 2001 *
- Oct 12, 2001 *
- Oct 13, 2001
- Oct 13, 2001 *

2002
27 pages

- Jan 06, 2002 *
- Jan 19, 2002 *
- Jan 25, 2002 *
- Feb 22, 2002 *
- May 26, 2002 *
- Jun 01, 2002 *
- Jun 04, 2002 *
- Jun 04, 2002 *
- Jul 26, 2002 *
- Aug 02, 2002 *
- Aug 22, 2002 *
- Sep 15, 2002 *
- Sep 25, 2002 *
- Sep 28, 2002 *
- Sep 30, 2002 *
- Oct 04, 2002 *
- Oct 10, 2002 *
- Oct 10, 2002 *
- Oct 14, 2002 *
- Oct 18, 2002 *
- Oct 31, 2002 *
- Nov 04, 2002 *
- Nov 09, 2002 *
- Nov 20, 2002 *
- Nov 26, 2002 *
- Nov 28, 2002 *
- Nov 29, 2002 *

2003
4 pages

- Jan 28, 2003 *
- Feb 03, 2003 *
- Feb 07, 2003 *
- Feb 13, 2003 *

Oct 13, 2001 *
Oct 14, 2001 *
Oct 14, 2001 *
Oct 15, 2001 *
Oct 15, 2001 *
Oct 16, 2001 *
Oct 16, 2001 *
Oct 18, 2001 *
Oct 18, 2001 *
Oct 19, 2001 *
Oct 19, 2001 *
Oct 20, 2001 *
Oct 20, 2001 *
Oct 21, 2001 *
Oct 21, 2001 *
Oct 22, 2001 *
Oct 22, 2001 *
Oct 23, 2001 *
Oct 24, 2001 *
Oct 25, 2001 *
Oct 26, 2001 *
Oct 27, 2001 *
Oct 27, 2001 *
Oct 28, 2001 *
Oct 30, 2001 *
Oct 30, 2001 *
Oct 31, 2001 *
Oct 31, 2001 *
Nov 01, 2001 *
Nov 01, 2001 *
Nov 01, 2001 *
Nov 02, 2001 *
Nov 02, 2001 *
Nov 02, 2001 *
Nov 02, 2001 *
Nov 03, 2001 *
Nov 04, 2001 *
Nov 05, 2001 *
Nov 05, 2001 *
Nov 05, 2001 *
Nov 05, 2001 *
Nov 06, 2001 *
Nov 06, 2001 *
Nov 06, 2001 *
Nov 07, 2001 *
Nov 08, 2001 *
Nov 08, 2001 *
Nov 09, 2001 *
Nov 09, 2001 *
Nov 10, 2001 *
Nov 10, 2001 *
Nov 11, 2001 *
Nov 12, 2001 *
Nov 12, 2001 *
Nov 13, 2001 *
Nov 14, 2001 *
Nov 14, 2001 *
Nov 14, 2001 *
Nov 15, 2001 *

Nov 17, 2001
Nov 18, 2001 *
Nov 19, 2001 *
Nov 20, 2001 *
Nov 21, 2001 *
Nov 21, 2001 *
Nov 22, 2001 *
Nov 22, 2001 *
Nov 23, 2001 *
Nov 24, 2001 *
Nov 24, 2001 *
Nov 25, 2001 *
Nov 26, 2001 *
Nov 26, 2001 *
Nov 28, 2001 *
Nov 29, 2001 *
Nov 29, 2001 *
Nov 30, 2001 *
Dec 01, 2001 *
Dec 01, 2001 *
Dec 02, 2001 *
Dec 03, 2001 *
Dec 04, 2001 *
Dec 05, 2001 *
Dec 06, 2001 *
Dec 07, 2001 *
Dec 08, 2001 *
Dec 10, 2001 *
Dec 11, 2001 *
Dec 12, 2001 *
Dec 13, 2001 *
Dec 14, 2001 *
Dec 15, 2001 *
Dec 16, 2001 *
Dec 16, 2001 *
Dec 17, 2001 *
Dec 17, 2001 *

Home | Help

INTERNET ARCHIVE
WaybackMachine

Enter Web Address: http:// All Take Me Back Adv. Search Compare Archive Pages

Searched for http://stamps.com **248** Results

Note some duplicates are not shown. See all.
* denotes when site was updated.

Search Results for Jan 01, 1996 - Dec 04, 2003

1996	1997	1998	1999	2000	2001	2002	2003
0 pages	0 pages	3 pages	5 pages	28 pages	143 pages	27 pages	4 pages
		Nov 11, 1998 *	Jan 25, 1999 *	Feb 29, 2000 *	Jan 18, 2001 *	Jan 06, 2002 *	Jan 28, 2003 *
		Dec 01, 1998 *	Apr 28, 1999 *	Mar 01, 2000	Jan 19, 2001	Jan 19, 2002 *	Feb 03, 2003 *
		Dec 12, 1998 *	Oct 01, 1999 *	Mar 03, 2000	Jan 27, 2001	Jan 25, 2002 *	Feb 07, 2003 *
			Oct 12, 1999	Mar 03, 2000 *	Feb 06, 2001	Feb 22, 2002 *	Feb 13, 2003 *
			Nov 17, 1999	Mar 04, 2000	Feb 26, 2001 *	May 26, 2002 *	
				Apr 07, 2000 *	Mar 01, 2001 *	Jun 01, 2002 *	
				May 10, 2000 *	Mar 02, 2001	Jun 04, 2002 *	
				May 10, 2000 *	Mar 09, 2001	Jun 04, 2002 *	
				May 11, 2000 *	Mar 30, 2001	Jul 26, 2002 *	
				May 12, 2000	Apr 05, 2001 *	Aug 02, 2002 *	
				May 20, 2000 *	Apr 29, 2001 *	Aug 22, 2002 *	
				May 20, 2000 *	May 05, 2001 *	Sep 15, 2002 *	
				Jun 13, 2000 *	May 07, 2001	Sep 25, 2002 *	
				Jun 19, 2000 *	May 12, 2001 *	Sep 28, 2002 *	
				Jun 20, 2000 *	May 16, 2001	Sep 30, 2002 *	
				Jun 21, 2000	May 17, 2001	Oct 04, 2002 *	
				Jun 22, 2000	May 18, 2001	Oct 10, 2002 *	
				Jul 06, 2000 *	May 19, 2001	Oct 10, 2002 *	
				Aug 15, 2000 *	May 22, 2001	Oct 14, 2002 *	
				Oct 06, 2000 *	May 31, 2001 *	Oct 18, 2002 *	
				Oct 08, 2000 *	Jun 01, 2001	Oct 31, 2002 *	
				Oct 08, 2000 *	Jun 02, 2001	Nov 04, 2002 *	
				Oct 17, 2000 *	Jun 03, 2001	Nov 09, 2002 *	
				Oct 18, 2000	Jun 06, 2001 *	Nov 20, 2002 *	
				Oct 19, 2000	Jun 07, 2001	Nov 26, 2002 *	
				Nov 09, 2000 *	Jun 11, 2001	Nov 28, 2002 *	
				Nov 19, 2000 *	Jun 14, 2001	Nov 29, 2002 *	
				Dec 02, 2000 *	Jun 26, 2001		
					Jun 27, 2001		
					Jun 29, 2001		
					Jun 30, 2001		
					Jul 01, 2001		
					Jul 08, 2001		
					Jul 10, 2001		
					Jul 13, 2001 *		
					Aug 25, 2001 *		
					Oct 09, 2001 *		
					Oct 09, 2001 *		
					Oct 09, 2001 *		
					Oct 10, 2001 *		
					Oct 10, 2001 *		
					Oct 11, 2001 *		
					Oct 11, 2001 *		
					Oct 12, 2001 *		
					Oct 12, 2001 *		
					Oct 13, 2001		
					Oct 13, 2001 *		

Oct 13, 2001 *
Oct 14, 2001 *
Oct 14, 2001 *
Oct 15, 2001 *
Oct 15, 2001 *
Oct 16, 2001 *
Oct 16, 2001 *
Oct 18, 2001 *
Oct 18, 2001 *
Oct 19, 2001 *
Oct 19, 2001 *
Oct 20, 2001 *
Oct 20, 2001 *
Oct 21, 2001 *
Oct 21, 2001 *
Oct 22, 2001 *
Oct 22, 2001 *
Oct 23, 2001 *
Oct 24, 2001 *
Oct 25, 2001 *
Oct 26, 2001 *
Oct 27, 2001 *
Oct 27, 2001 *
Oct 28, 2001 *
Oct 30, 2001 *
Oct 30, 2001 *
Oct 31, 2001 *
Oct 31, 2001 *
Nov 01, 2001 *
Nov 01, 2001 *
Nov 01, 2001 *
Nov 02, 2001 *
Nov 02, 2001 *
Nov 02, 2001 *
Nov 02, 2001 *
Nov 03, 2001 *
Nov 04, 2001 *
Nov 05, 2001 *
Nov 05, 2001 *
Nov 05, 2001 *
Nov 05, 2001 *
Nov 06, 2001 *
Nov 06, 2001 *
Nov 06, 2001 *
Nov 07, 2001 *
Nov 08, 2001 *
Nov 08, 2001 *
Nov 09, 2001 *
Nov 09, 2001 *
Nov 10, 2001 *
Nov 10, 2001 *
Nov 11, 2001 *
Nov 12, 2001 *
Nov 12, 2001 *
Nov 13, 2001 *
Nov 14, 2001 *
Nov 14, 2001 *
Nov 14, 2001 *
Nov 15, 2001 *

Nov 17, 2001
Nov 18, 2001 *
Nov 19, 2001 *
Nov 20, 2001 *
Nov 21, 2001 *
Nov 21, 2001 *
Nov 22, 2001 *
Nov 22, 2001 *
Nov 23, 2001 *
Nov 24, 2001 *
Nov 24, 2001 *
Nov 25, 2001 *
Nov 26, 2001 *
Nov 26, 2001 *
Nov 28, 2001 *
Nov 29, 2001 *
Nov 29, 2001 *
Nov 30, 2001 *
Dec 01, 2001 *
Dec 01, 2001 *
Dec 02, 2001 *
Dec 03, 2001 *
Dec 04, 2001 *
Dec 05, 2001 *
Dec 06, 2001 *
Dec 07, 2001 *
Dec 08, 2001 *
Dec 10, 2001 *
Dec 11, 2001 *
Dec 12, 2001 *
Dec 13, 2001 *
Dec 14, 2001 *
Dec 15, 2001 *
Dec 16, 2001 *
Dec 16, 2001 *
Dec 17, 2001 *
Dec 17, 2001 *

Home | Help

INTERNET ARCHIVE
WaybackMachine

Enter Web Address: http:// All Take Me Back Adv. Search Compare Archive Pages

Searched for http://www.stamps.com **248** Results

Note some duplicates are not shown. See all.
* denotes when site was updated.

Search Results for Jan 01, 1996 - Dec 04, 2003

1996	1997	1998	1999	2000	2001	2002	2003
0 pages	0 pages	3 pages	5 pages	28 pages	143 pages	27 pages	4 pages
		Nov 11, 1998 *	Jan 25, 1999 *	Feb 29, 2000 *	Jan 18, 2001 *	Jan 06, 2002 *	Jan 28, 2003 *
		Dec 01, 1998 *	Apr 28, 1999 *	Mar 01, 2000	Jan 19, 2001	Jan 19, 2002 *	Feb 03, 2003 *
		Dec 12, 1998 *	Oct 01, 1999 *	Mar 03, 2000	Jan 27, 2001	Jan 25, 2002 *	Feb 07, 2003 *
			Oct 12, 1999	Mar 03, 2000 *	Feb 06, 2001	Feb 22, 2002 *	Feb 13, 2003 *
			Nov 17, 1999	Mar 04, 2000	Feb 26, 2001 *	May 26, 2002 *	
				Apr 07, 2000 *	Mar 01, 2001 *	Jun 01, 2002 *	
				May 10, 2000 *	Mar 02, 2001	Jun 04, 2002 *	
				May 10, 2000 *	Mar 09, 2001	Jun 04, 2002 *	
				May 11, 2000 *	Mar 30, 2001	Jul 26, 2002 *	
				May 12, 2000	Apr 05, 2001 *	Aug 02, 2002 *	
				May 20, 2000 *	Apr 29, 2001 *	Aug 22, 2002 *	
				May 20, 2000 *	May 05, 2001 *	Sep 15, 2002 *	
				Jun 13, 2000 *	May 07, 2001	Sep 25, 2002 *	
				Jun 19, 2000 *	May 12, 2001 *	Sep 28, 2002 *	
				Jun 20, 2000 *	May 16, 2001	Sep 30, 2002 *	
				Jun 21, 2000	May 17, 2001	Oct 04, 2002 *	
				Jun 22, 2000	May 18, 2001	Oct 10, 2002 *	
				Jul 06, 2000 *	May 19, 2001	Oct 10, 2002 *	
				Aug 15, 2000 *	May 22, 2001	Oct 14, 2002 *	
				Oct 06, 2000 *	May 31, 2001 *	Oct 18, 2002 *	
				Oct 08, 2000 *	Jun 01, 2001	Oct 31, 2002 *	
				Oct 08, 2000 *	Jun 02, 2001	Nov 04, 2002 *	
				Oct 17, 2000 *	Jun 03, 2001	Nov 09, 2002 *	
				Oct 18, 2000	Jun 06, 2001 *	Nov 20, 2002 *	
				Oct 19, 2000	Jun 07, 2001	Nov 26, 2002 *	
				Nov 09, 2000 *	Jun 11, 2001	Nov 28, 2002 *	
				Nov 19, 2000 *	Jun 14, 2001	Nov 29, 2002 *	
				Dec 02, 2000 *	Jun 26, 2001		
					Jun 27, 2001		
					Jun 29, 2001		
					Jun 30, 2001		
					Jul 01, 2001		
					Jul 08, 2001		
					Jul 10, 2001		
					Jul 13, 2001 *		
					Aug 25, 2001 *		
					Oct 09, 2001 *		
					Oct 09, 2001 *		
					Oct 09, 2001 *		
					Oct 10, 2001 *		
					Oct 10, 2001 *		
					Oct 11, 2001 *		
					Oct 11, 2001 *		
					Oct 12, 2001 *		
					Oct 12, 2001 *		
					Oct 13, 2001		
					Oct 13, 2001 *		

Oct 13, 2001 *
Oct 14, 2001 *
Oct 14, 2001 *
Oct 15, 2001 *
Oct 15, 2001 *
Oct 16, 2001 *
Oct 16, 2001 *
Oct 18, 2001 *
Oct 18, 2001 *
Oct 19, 2001 *
Oct 19, 2001 *
Oct 20, 2001 *
Oct 20, 2001 *
Oct 21, 2001 *
Oct 21, 2001 *
Oct 22, 2001 *
Oct 22, 2001 *
Oct 23, 2001 *
Oct 24, 2001 *
Oct 25, 2001 *
Oct 26, 2001 *
Oct 27, 2001 *
Oct 27, 2001 *
Oct 28, 2001 *
Oct 30, 2001 *
Oct 30, 2001 *
Oct 31, 2001 *
Oct 31, 2001 *
Nov 01, 2001 *
Nov 01, 2001 *
Nov 01, 2001 *
Nov 02, 2001 *
Nov 02, 2001 *
Nov 02, 2001 *
Nov 02, 2001 *
Nov 03, 2001 *
Nov 04, 2001 *
Nov 05, 2001 *
Nov 05, 2001 *
Nov 05, 2001 *
Nov 05, 2001 *
Nov 06, 2001 *
Nov 06, 2001 *
Nov 06, 2001 *
Nov 07, 2001 *
Nov 08, 2001 *
Nov 08, 2001 *
Nov 09, 2001 *
Nov 09, 2001 *
Nov 10, 2001 *
Nov 10, 2001 *
Nov 11, 2001 *
Nov 12, 2001 *
Nov 12, 2001 *
Nov 13, 2001 *
Nov 14, 2001 *
Nov 14, 2001 *
Nov 14, 2001 *
Nov 15, 2001 *

Nov 17, 2001
Nov 18, 2001 *
Nov 19, 2001 *
Nov 20, 2001 *
Nov 21, 2001 *
Nov 21, 2001 *
Nov 22, 2001 *
Nov 22, 2001 *
Nov 23, 2001 *
Nov 24, 2001 *
Nov 24, 2001 *
Nov 25, 2001 *
Nov 26, 2001 *
Nov 26, 2001 *
Nov 28, 2001 *
Nov 29, 2001 *
Nov 29, 2001 *
Nov 30, 2001 *
Dec 01, 2001 *
Dec 01, 2001 *
Dec 02, 2001 *
Dec 03, 2001 *
Dec 04, 2001 *
Dec 05, 2001 *
Dec 06, 2001 *
Dec 07, 2001 *
Dec 08, 2001 *
Dec 10, 2001 *
Dec 11, 2001 *
Dec 12, 2001 *
Dec 13, 2001 *
Dec 14, 2001 *
Dec 15, 2001 *
Dec 16, 2001 *
Dec 16, 2001 *
Dec 17, 2001 *
Dec 17, 2001 *

Home | Help

INTERNET ARCHIVE
WaybackMachine

Enter Web Address: http:// All Take Me Back Adv. Search Compare Archive Pages

Searched for http://www.endicia.com/ **54** Results

Note some duplicates are not shown. See all.
* denotes when site was updated.

Search Results for Jan 01, 1996 - Dec 04, 2003

1996	1997	1998	1999	2000	2001	2002	2003
0 pages	0 pages	0 pages	0 pages	0 pages	8 pages	43 pages	1 pages
					Feb 02, 2001 *	May 25, 2002 *	Feb 10, 2003 *
					Feb 03, 2001 *	May 31, 2002	
					Mar 02, 2001 *	Jul 21, 2002	
					Mar 02, 2001 *	Aug 02, 2002	
					Mar 31, 2001 *	Sep 21, 2002 *	
					Apr 05, 2001 *	Sep 24, 2002	
					Apr 21, 2001 *	Sep 26, 2002	
					May 16, 2001 *	Sep 27, 2002	
						Oct 16, 2002	
						Oct 17, 2002	
						Oct 18, 2002	
						Oct 19, 2002	
						Oct 20, 2002	
						Oct 21, 2002	
						Oct 22, 2002	
						Oct 23, 2002	
						Oct 24, 2002	
						Oct 25, 2002	
						Oct 26, 2002	
						Oct 27, 2002	
						Oct 28, 2002	
						Oct 29, 2002	
						Oct 30, 2002	
						Oct 31, 2002	
						Nov 02, 2002	
						Nov 03, 2002	
						Nov 04, 2002	
						Nov 05, 2002	
						Nov 06, 2002	
						Nov 07, 2002	
						Nov 08, 2002	
						Nov 09, 2002	
						Nov 11, 2002	
						Nov 13, 2002	
						Nov 14, 2002	
						Nov 16, 2002	
						Nov 17, 2002	
						Nov 18, 2002	
						Nov 19, 2002	
						Nov 23, 2002	
						Nov 24, 2002	
						Nov 29, 2002	
						Dec 02, 2002	

Home | Help

INTERNET ARCHIVE
WaybackMachine

Enter Web Address: http:// All Take Me Back Adv. Search

Robots.txt Query Exclusion.

We're sorry, access to http://www.estamp.com has been blocked by the site owner via robots.txt.
Read more about robots.txt
See the site's robots.txt file.
Try another request or click here to search for all pages on estamp.com/
See the FAQs for more info and help, or contact us.

Home | Help

INTERNET ARCHIVE
WaybackMachine

Enter Web Address: http:// All Take Me Back Adv. Search

Searched for http://www.labelcity.com/dymointernetpostagelabel.shtml **3** Results

* denotes when site was updated.

Search Results for Jan 01, 1996 - Dec 04, 2003

1996	1997	1998	1999	2000	2001	2002	2003
0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	2 pages	1 pages
						Jun 12, 2002 * Aug 11, 2002	Mar 01, 2003

Home | Help

INTERNET ARCHIVE
WaybackMachine

Enter Web Address: http:// All Take Me Back Adv. Search Compare Archive Pages

Searched for **http://www.labelcity.com/**

41 Results

Note some duplicates are not shown. See all.
* denotes when site was updated.

Search Results for Jan 01, 1996 - Dec 04, 2003

1996	1997	1998	1999	2000	2001	2002	2003
0 pages	0 pages	0 pages	2 pages	10 pages	7 pages	17 pages	4 pages
			Oct 22, 1999 *	Mar 03, 2000 *	Feb 03, 2001	Jan 23, 2002	Feb 02, 2003
			Nov 03, 1999 *	Mar 04, 2000	Apr 02, 2001 *	Jan 24, 2002	Feb 08, 2003
				May 10, 2000 *	Apr 05, 2001	May 27, 2002 *	Feb 13, 2003
				May 20, 2000 *	May 16, 2001 *	May 28, 2002	Feb 14, 2003
				Jun 18, 2000 *	Jul 20, 2001 *	May 30, 2002	
				Jun 21, 2000	Sep 27, 2001 *	Jun 03, 2002	
				Aug 17, 2000 *	Nov 27, 2001	Jul 03, 2002 *	
				Oct 14, 2000 *		Jul 20, 2002 *	
				Oct 17, 2000		Jul 22, 2002	
				Nov 09, 2000		Aug 02, 2002	
						Sep 06, 2002 *	
						Sep 22, 2002 *	
						Sep 25, 2002	
						Sep 29, 2002	
						Nov 21, 2002	
						Nov 22, 2002	
						Nov 24, 2002	

Home | Help

Patent Assignment Abstract of Title

Total Assignments: 1

Application #: 09684861 **Filing Dt:** 10/06/2000 **Patent #:** NONE **Issue Dt:**

PCT #: NONE **Publication #:** NONE **Pub Dt:**

Inventors: Paul Bilibin, Jinyue Liu

Title: Apparatus, systems and methods for determining delivery time schedules for each of multiple carriers

Assignment: 1

Reel/Frame: 011630/0196 **Received:** 04/02/2001 **Recorded:** 03/13/2001 **Mailed:** 06/05/2001 **Pages:** 3

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors: BILIBIN, PAUL **Exec Dt:** 01/23/2001

LIU, JINYUE **Exec Dt:** 12/06/2000

Assignee: STAMPS.COM INC.
3420 OCEAN PARK BLVD., SUITE 1040
SANTA MONICA, CALIFORNIA 90405

Correspondent: KHORSANDI PATENT LAW GROUP A LAW CORP.
MARILYN R. KHORSANDI
140 S. LAKE AVE., SUITE 312
PASADENA, CA 91101-4710

Search Results as of: 12/4/2003 11:12:05 A.M.

If you have any comments or questions concerning the data displayed, contact OPR / Assignments at 703-308-9723
Web interface last modified: Oct. 5, 2002